

Vorlesung „Embedded Software-Engineering im Bereich Automotive“

Technische Universität Dresden, Fakultät Informatik,
Professur Softwaretechnologie

Sommersemester 2010

Dr. rer. nat. Bernhard Hohlfeld

bernhard.hohlfeld@daad-alumni.de

1. Motivation und Überblick

2. Grundlagen Fahrzeugentwicklung, KFZ-Elektronik und Software

3. Übersicht Automotive Elektrik/Elektronik-Entwicklung (E/E)

4. Kernprozess zur Entwicklung von elektronischen Systemen und Software

5. Unterstützungsprozesse für die Embedded Software Entwicklung

6. Beispiele aus der Praxis

7. Wichtige Normen/Standards/Empfehlungen für die Embedded Software Entwicklung

1. Motivation und Überblick

1. Bedeutung der Automobilindustrie für die deutsche Volkswirtschaft
2. Motivation Automotive Software Engineering
3. GI-Fachgruppe Automotive Software Engineering
4. Software Entwicklungsprozess
5. Standards zur Softwareentwicklung
6. Software Architekturen
7. AUTOSAR
8. Lessons Learned

1. Motivation und Überblick



1. Bedeutung der Automobilindustrie für die deutsche Volkswirtschaft

2. Motivation Automotive Software Engineering

3. GI-Fachgruppe Automotive Software Engineering

4. Software Entwicklungsprozess

5. Standards zur Softwareentwicklung

6. Software Architekturen

7. AUTOSAR

8. Lessons Learned

1. Bedeutung der Automobilindustrie für die deutsche Volkswirtschaft



■ Beispiele

- Daimler Nutzfahrzeuge
- Mercedes Benz PKW
- BMW Group

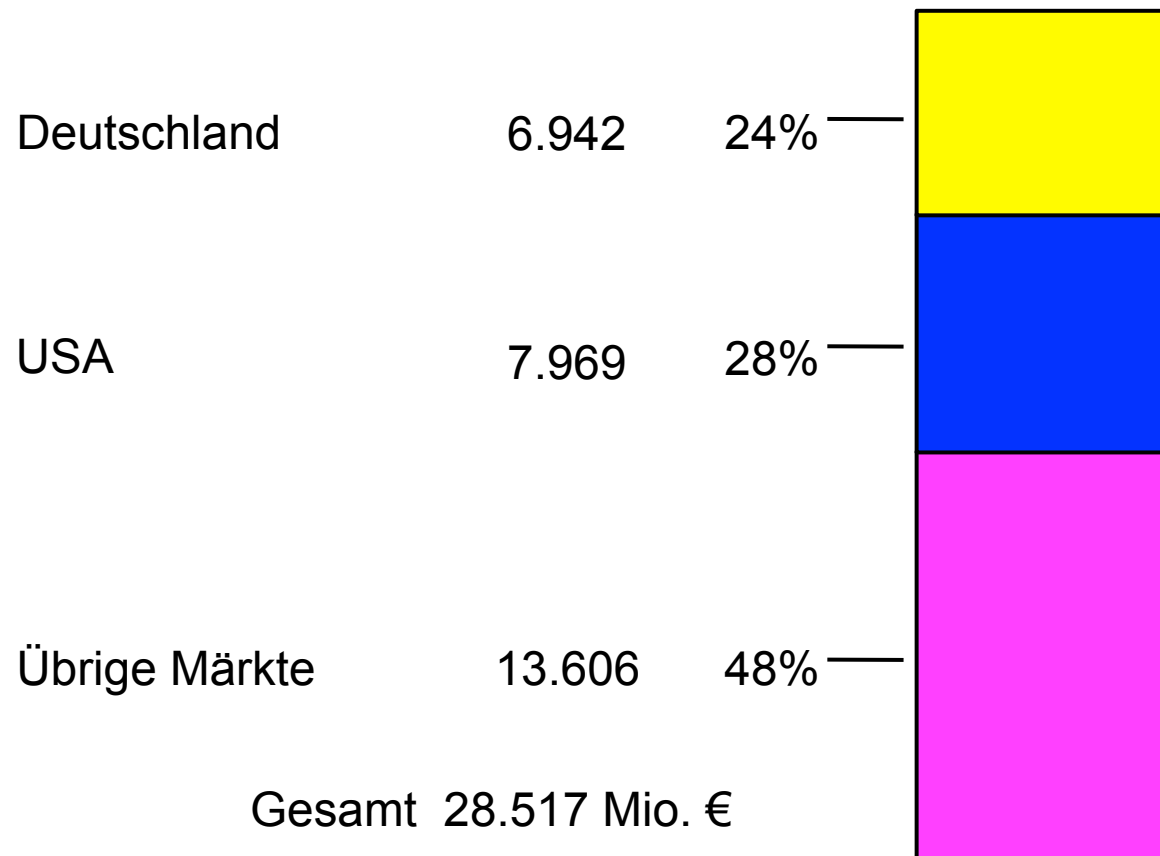
■ Die Daten stammen aus den Jahren 2003 und 2004

■ Sie gelten aber in der Tendenz bis Mitte 2008

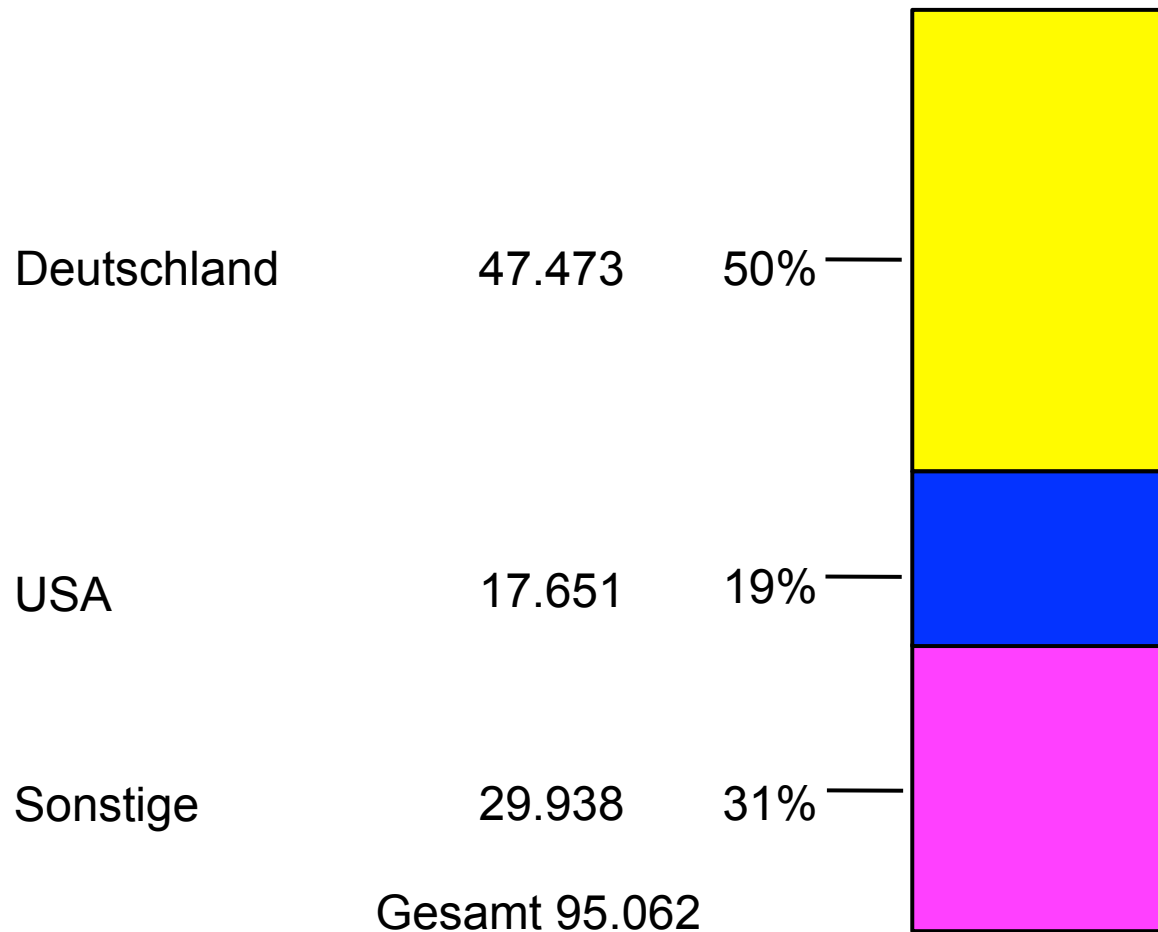
■ Einige Relationen gelten auch noch 2009

■ Auswirkungen der Finanzkrise

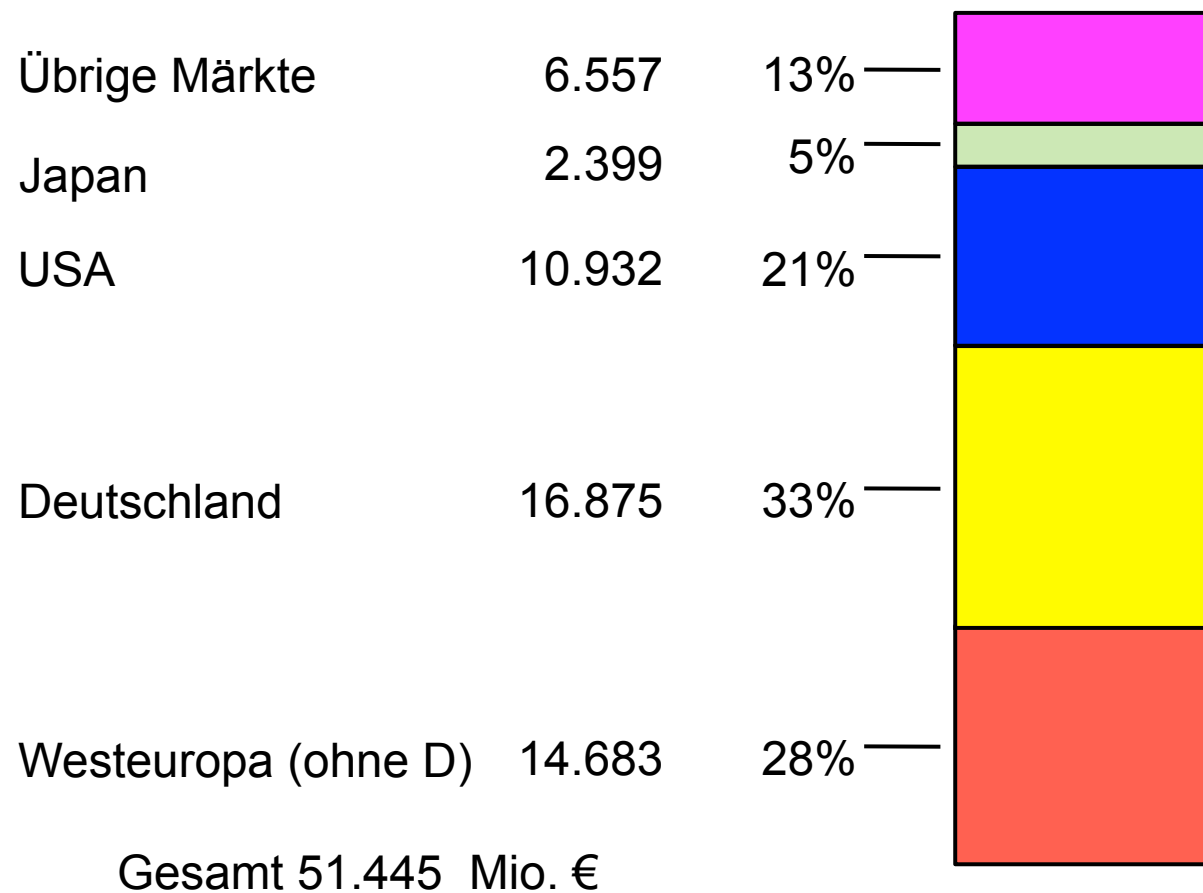
Daimler Nutzfahrzeuge Umsatz 2003



Daimler Nutzfahrzeuge Mitarbeiter 31.12. 2003



Mercedes Benz PKW Umsatz 2003



Mercedes Benz PKW Mitarbeiter 31.12. 2003



Übrige Länder 8.204 8%

USA 2.191 2%

Deutschland 93.756 90%

Gesamt 104.151



Zum Vergleich: BMW-Group im Jahr 2004



■ BMW Group wichtigste Automobilmärkte 2004 in % vom Absatz

■ USA	24,5%
■ Deutschland	23,5%
■ Grossbritannien	12,0%
■ Rest	40,0%

■ Automobilproduktion der BMW Group im Jahr 2004

■ Dingolfing	304.300
■ Regensburg	262.500
■ München	172.200
■ Deutschland	739.000 (59%)
■ Ausserhalb Deutschland	511.500 (41%)

■ Quelle: Geschäftsbericht 2004 (Kurzfassung)



Auswirkungen der Finanzkrise Daimler LKW



Daimler LKW	2003	2008	2009	
Umsatz	28517	28572	18360	
Beschäftigte	95062	79415	70699	
Deutschland	47.473	30060	28582	40 %
USA	17.651	15004	11026	16 %
Sonstige	29.938	34351	31091	44 %
Absatz (Einheiten)		472100	259300	
Deutschland		41600	25000	10 %
Westeuropa ohne D		45100	19300	7 %
USA		78000	52400	20 %
Asien		164800	93000	36 %
Sonstige		142600	69600	27 %

Auswirkungen der Finanzkrise Mercedes Benz PKW



Mercedes Benz PKW	2003	2008	2009	
Umsatz	51445	47772	41318	
Beschäftigte	104151	97303	93572	
Deutschland	93756	85046	83156	89 %
USA	2191	3782	2992	3 %
ROW	8204	8475	7424	8 %
Absatz (Einheiten)		1273000	1093900	
Deutschland		332500	297800	27 %
Westeuropa ohne D		400700	325700	30 %
USA		251200	203000	19 %
NAFTA ohne USA		31000	32500	3 %
China		48600	67500	6 %
Japan		37000	26700	2 %
Asien/Pazifik ohne Ch, J		73100	61700	6 %
Sonstige		98900	79000	7 %

1. Motivation und Überblick

1. Bedeutung der Automobilindustrie für die deutsche Volkswirtschaft

2. Motivation Automotive Software Engineering

3. GI-Fachgruppe Automotive Software Engineering

4. Software Entwicklungsprozess

5. Standards zur Softwareentwicklung

6. Software Architekturen

7. AUTOSAR

8. Lessons Learned

A nice car



Source: Dr. Michael Reinfrank, Siemens VDO Automotive

A powerful engine



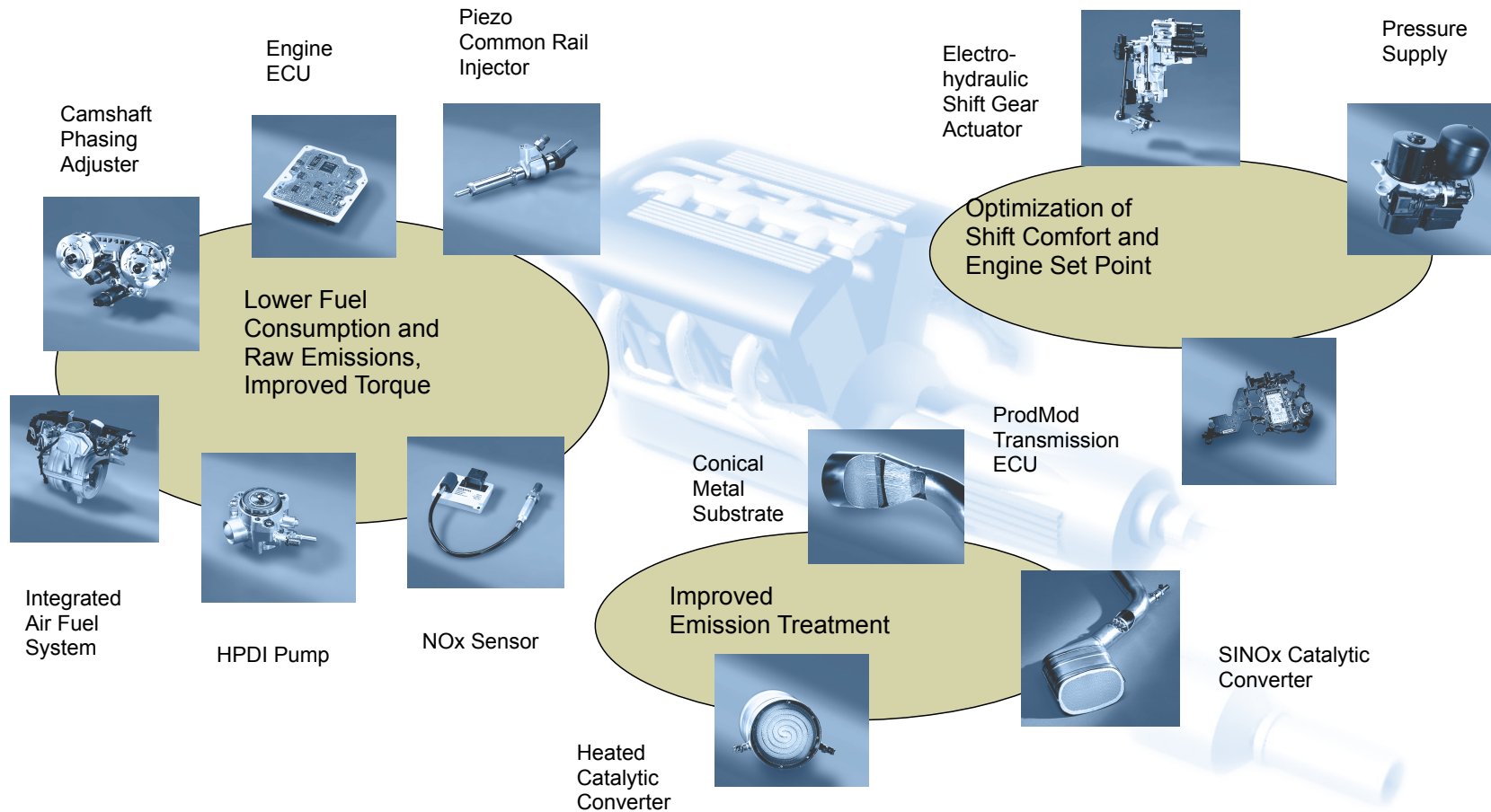
Source: Dr. Michael Reinfrank, Siemens VDO Automotive

A look inside



Source: Dr. Michael Reinfrank, Siemens VDO Automotive

Software is the intelligent “glue” for systems



Source: Dr. Michael Reinfrank, Siemens VDO Automotive

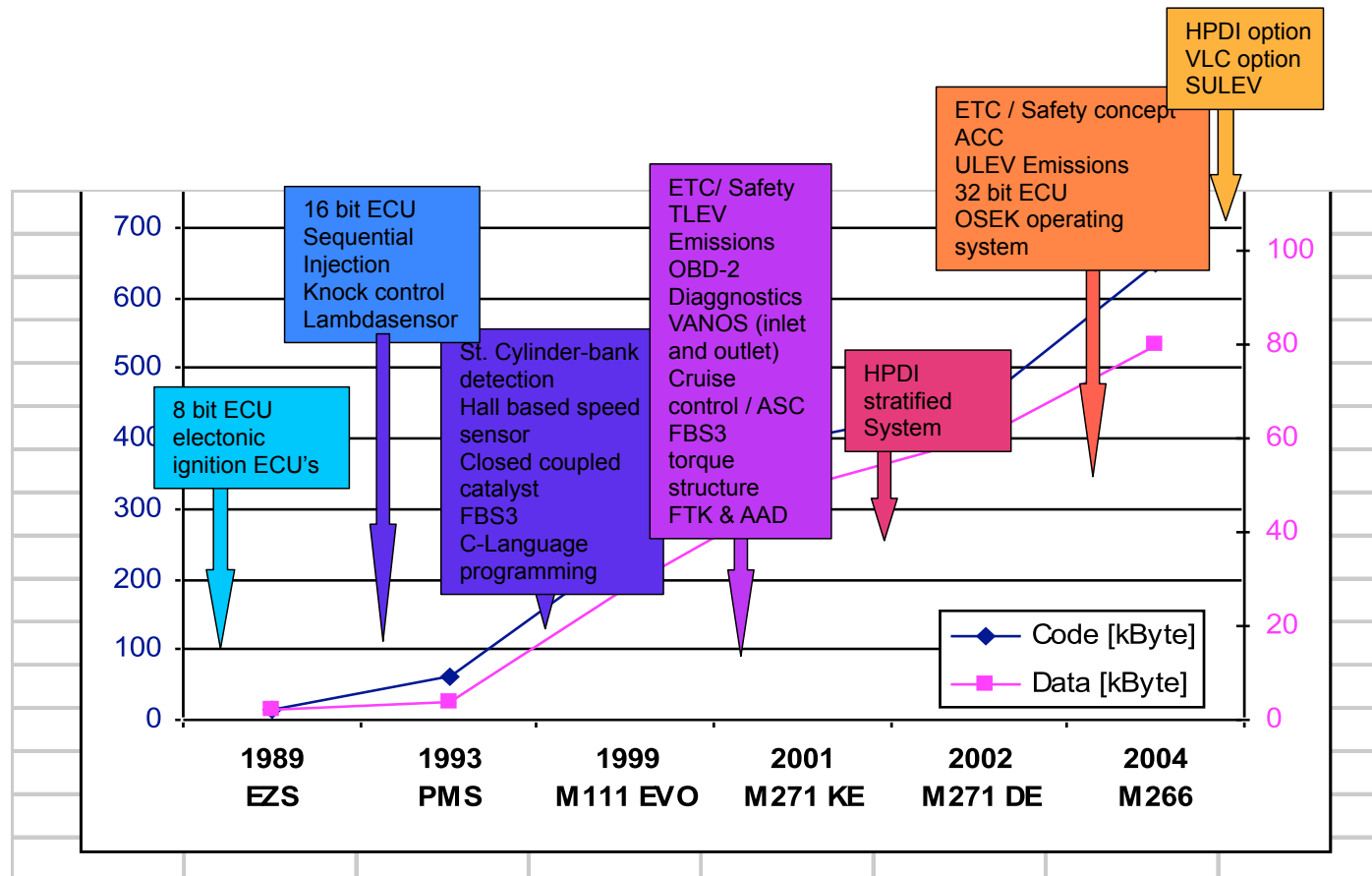
The intelligence inside: 5000 pages Software



<pre>#define MEM_HINT MEM_HINT_SPEED_NORMAL_CODE4 #include <gmempi.h> void c_ef_act_100ms(void) /* called every 100 ms */ //***** // Calculation of LV_EF every 100ms //***** u8 tmp_ef_trh; if <lv_ef_ext_adj> then lv_ef = lv_act_ef_ext_adj;4 else if <lv_var_ef> then if <lv_st_end> then //***** Conditions fulfilled //***** if < t_ast <= c_t_ef_ast > then lv_ef = 0; if < lv_ch_n_sp_is > then lv_ef = c_state_ch_ef; if <lv_at> then if <vs < c_vs_min_ef_at> then lv_ef = c_state_vs_m_in_ef_a_t;4 Determination of the pv-threshold Determination of lv_ef depending on pedal-value else if <vs < c_vs_min_ef_mt> then lv_ef = c_state_vs_m_in_ef_m_t;4 Determination of the pv-threshold Determination of lv_ef depending on pedal-value else //***** Conditions not fulfilled //***** else //***** no ef configured //***** else TRL_DOP_PinLevel<EF_1, lv_ef>; /* switch pin */ else TRL_DOP_PinLevel<EF_1, lv_ef>; /* switch pin */ else TRL_DOP_PinLevel<EF_1, lv_ef>; /* switch pin */ else TRL_DOP_PinLevel<EF_1, lv_ef>; /* switch pin */ else TRL_DOP_PinLevel<EF_1, lv_ef>; /* switch pin */ else TRL_DOP_PinLevel<EF_1, lv_ef>; /* switch pin */ </pre>			
---	--	--	--

Source: Dr. Michael Reinfrank, Siemens VDO Automotive

Add code to add functions



Source: Dr. Michael Reinfrank, Siemens VDO Automotive

Heutige Fahrzeuge haben teilweise mehr als 50 Steuergeräte, die weit über 500.000 Zeilen Code enthalten. Über bis zu vier verschiedene Kommunikationsbusse gehen hunderte von Nachrichten und tausende von Signalen. Über zwei Drittel aller Innovationen im Automobil sind schon heute softwarebasiert, ein Anstieg der Softwareentwicklungskosten an den gesamten Entwicklungskosten von derzeit ca. 4% auf über 10% wird prognostiziert. Ein Automobil bündelt so auf 5x2m viele Fragestellungen der Informatik, insbesondere der Entwicklung komplexer und zuverlässiger Softwaresysteme.

Die Vorlesung führt in die Grundlagen und Besonderheiten des Software-Engineerings für elektronische Systeme im Automobil ein:

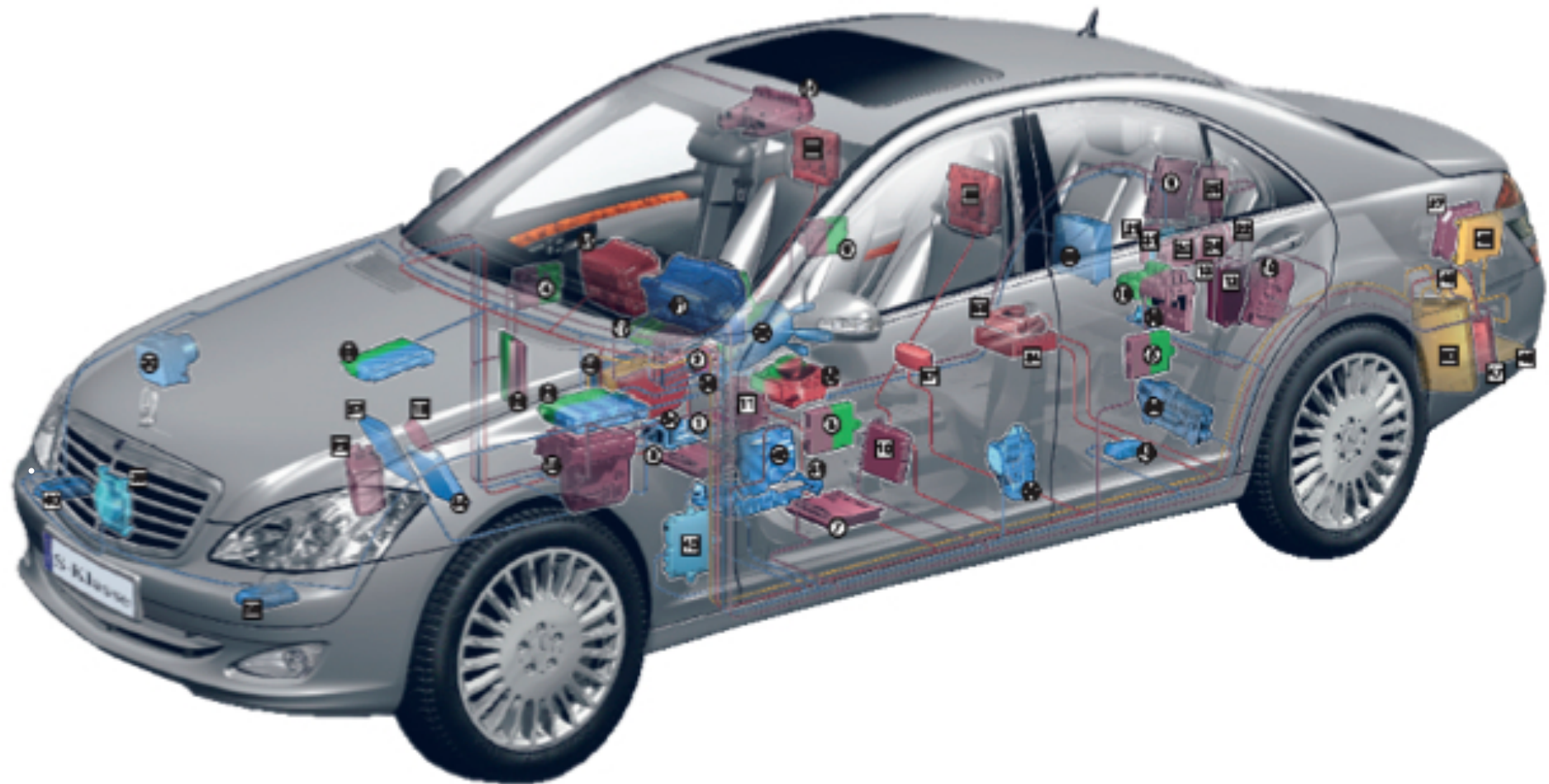
- Verteilte und komplexe Systementwicklung zwischen OEM und Zulieferern
- Sehr hohe Anforderungen an Zuverlässigkeit, Sicherheit (Safety und Security) sowie Echtzeitverhalten
- Extreme Umweltbedingungen (mechanische Beanspruchung, Verbauraum, Temperatur, etc.)
- Unterschiedliche Entwicklungs- und Lebenszyklen zwischen Produkt (Fahrzeug) und Software (Komponente)
- Hoher Zeit- und Kostendruck mit vielen Änderungs- und Konfigurationsanforderungen

Heutige Fahrzeuge haben teilweise mehr als 50 Steuergeräte, die weit über 500.000 Zeilen Code enthalten. Über bis zu vier verschiedene Kommunikationsbusse gehen hunderte von Nachrichten und tausende von Signalen. Über zwei Drittel aller Innovationen im Automobil sind schon heute softwarebasiert, ein Anstieg der Softwareentwicklungskosten an den gesamten Entwicklungskosten von derzeit ca. 4% auf über 10% wird prognostiziert. Ein Automobil bündelt so auf 5x2m viele Fragestellungen der Informatik, insbesondere der Entwicklung komplexer und zuverlässiger Softwaresysteme.

Die Vorlesung führt in die Grundlagen und Besonderheiten des Software-Engineerings für elektronische Systeme im Automobil ein:

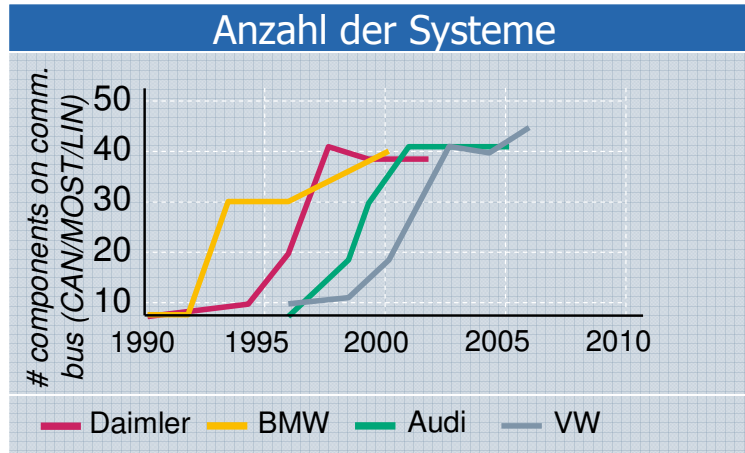
- Verteilte und komplexe Systementwicklung zwischen OEM und Zulieferern
- Sehr hohe Anforderungen an Zuverlässigkeit, Sicherheit (Safety und Security) sowie Echtzeitverhalten
- Extreme Umweltbedingungen (mechanische Beanspruchung, Verbauraum, Temperatur, etc.)
- Unterschiedliche Entwicklungs- und Lebenszyklen zwischen Produkt (Fahrzeug) und Software (Komponente)
- Hoher Zeit- und Kostendruck mit vielen Änderungs- und Konfigurationsanforderungen

Steigender Software Anteil

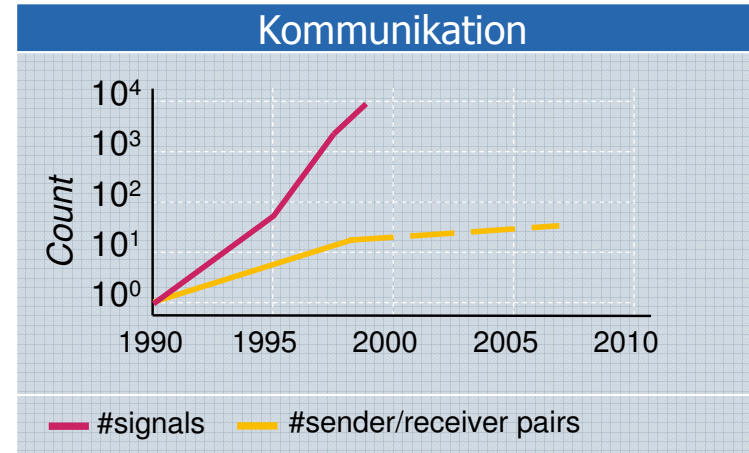


Komplexitätskrise

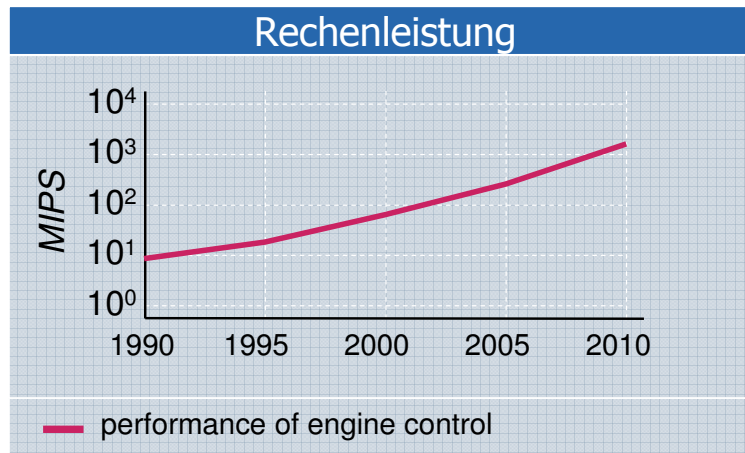
Die Komplexität von E/E-Systemen im Automobil wächst zunehmend



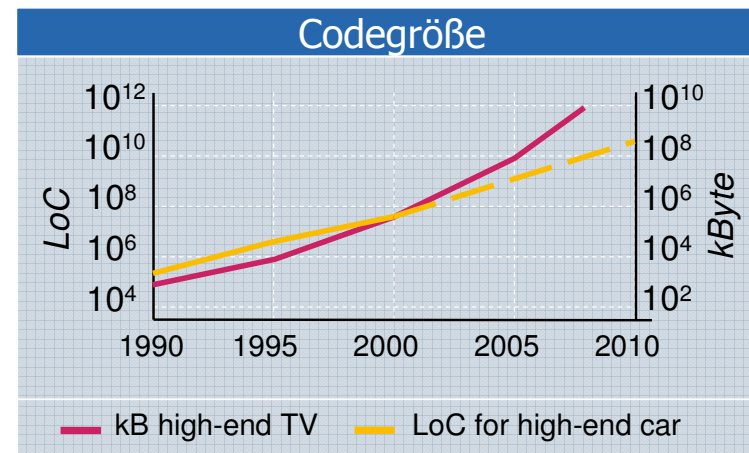
Quelle: VW 2005, Fachkongreß Automobil-Elektronik



Quelle: BMW, Frischkorn, BoCSE 2002



Quelle: NEC, 2006 (TOP57)



Quellen: Daimler-Chrysler 2004; Philips

Extreme Umweltbedingungen (mechanische Beanspruchung, Verbaauraum, Temperatur, etc.)



■ Was hat das mit Softwareentwicklung zu tun?



Unterschiedliche Entwicklungs- und Lebenszyklen zwischen Produkt (Fahrzeug) und Software (Komponente)



Unterschiedliche Entwicklungs- und Lebenszyklen zwischen Produkt (Fahrzeug) und Software (Komponente)



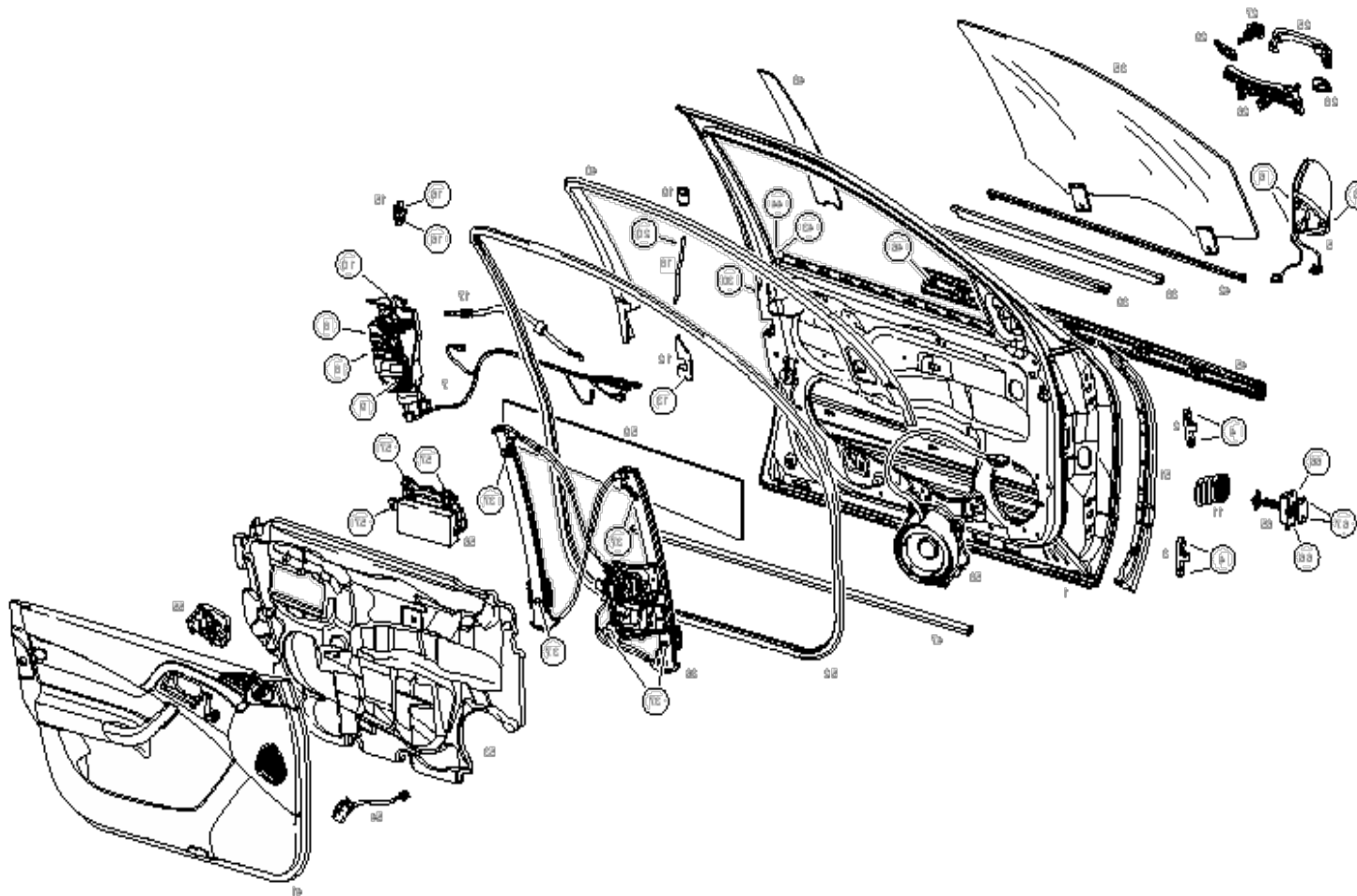
Hoher Zeit- und Kostendruck mit vielen Änderungs- und Konfigurationsanforderungen





Umfangreiche
Bedienungsanleitungen

Verteilte und komplexe Systementwicklung zwischen OEM und Zulieferern: Beispiel Türsteuerung



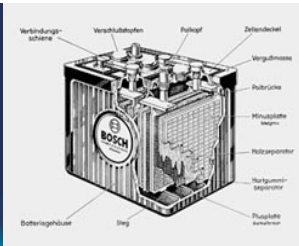
Automotive Software Engineering Historie



1897
Bosch
Magnet Ignition



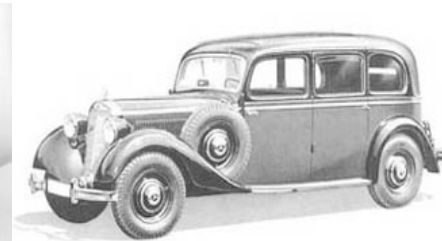
1913
Lighting System



1922/27
Battery



1930/32:
Radio



1936: Diesel Injection



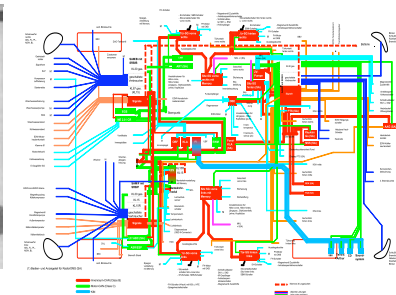
1950: Direct
Gasoline Injection



1967: First Electronic
Gasoline Injection



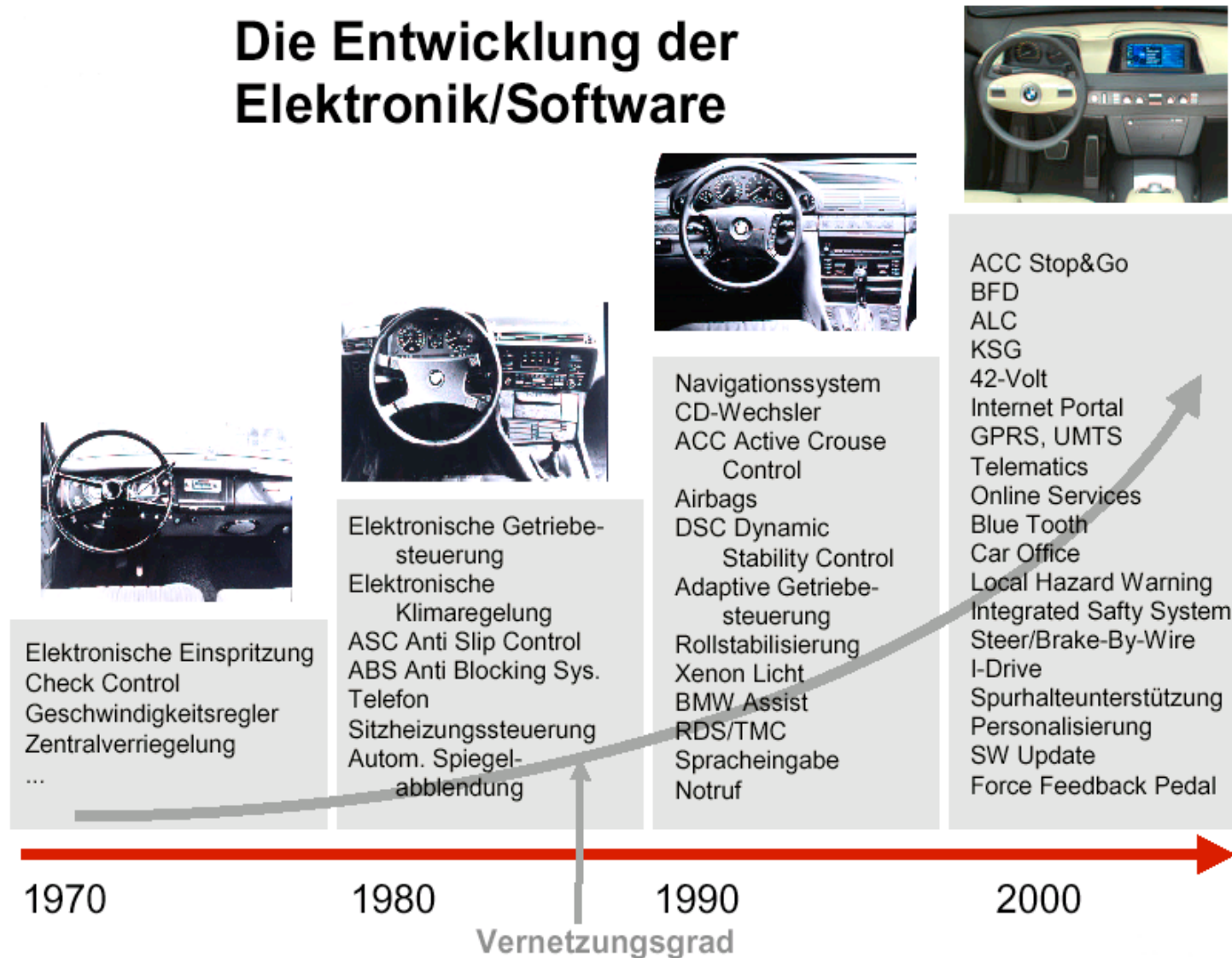
1979: First Integrated Engine
Management System (BMW
732i)



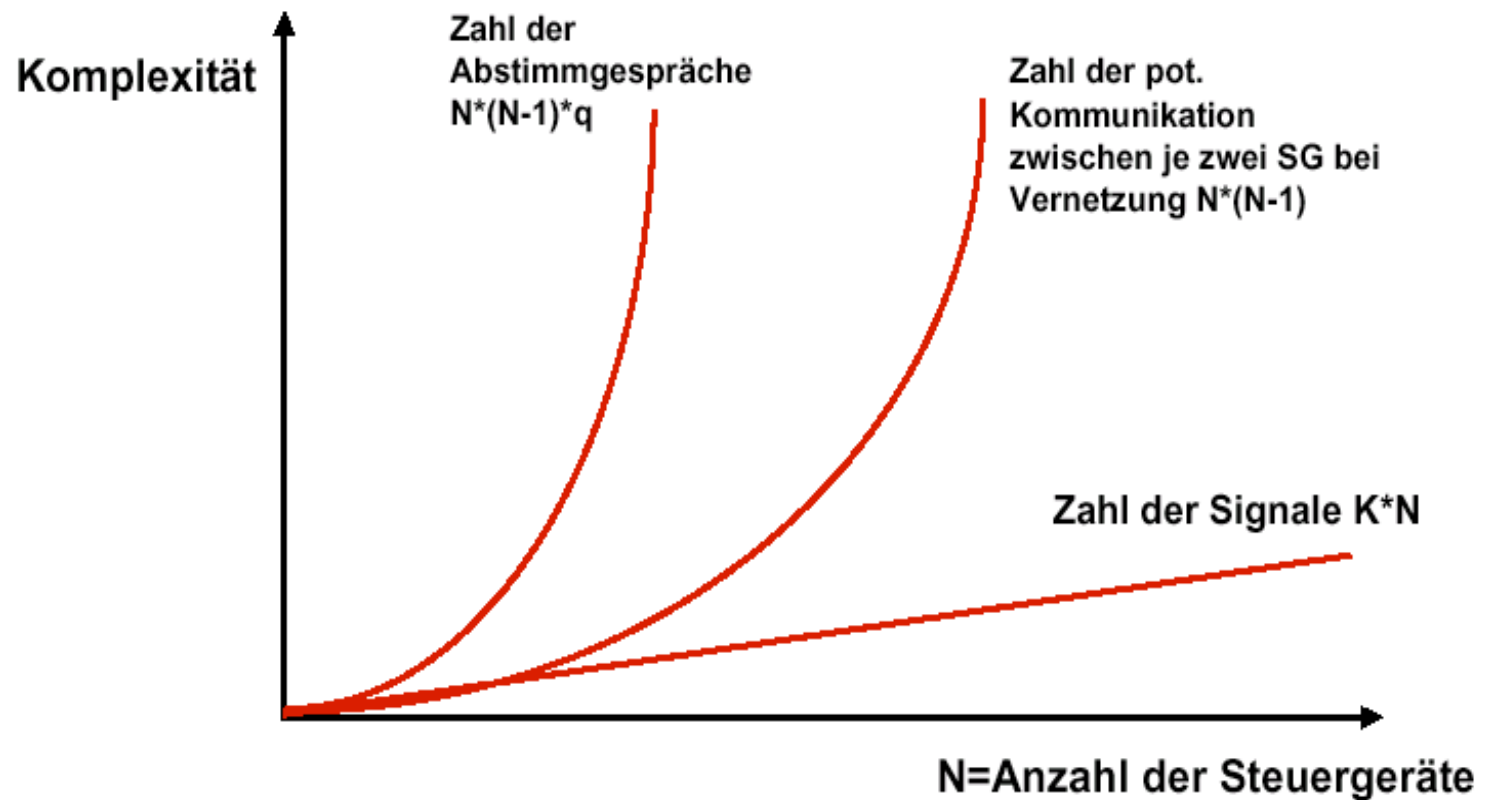
Networking
1991 CAN Bus System
2001 MOST

Kfz-Elektroniksysteme waren schon immer verteilt, jetzt sind sie zudem vernetzt

Die Entwicklung der Elektronik/Software



Entwicklung der Komplexität



Türsteuerung: Standard bis ca. 1990



■ Funktion “Tür entriegeln”

- Von aussen über Schlüssel
- Von innen über Hebel

■ Funktion “Tür verriegeln”

- Von aussen über Schlüssel
- Von innen über Knopf

■ Komfortfunktion

- Einzeltür / Zentral für alle Türen
- Kindersicherung

■ Realisierung der Funktionen über Mechanik

■ Kein SW-Anteil

Baugruppenverantwortlicher Türe (bis ca. 1990)



■ Ansprechpartner

- Baugruppenverantwortlicher Karosserie

■ Zulieferer

- Schliesssystem
- Scheiben
- Fensterheber
- Aussenspiegel

■ Schnittstellen

- Mechanik

■ Vorgaben

- Funktionalität
- Qualität
- Zeitziel
- Kostenziel
- Gewichtsziel
- Bauraum

Türsteuerung: Standard heute (Mittelklasse)



■ Funktion "Tür entriegeln"

- Von aussen über Schlüssel (Mechanik)
- Von aussen über Funkschlüssel (Mechatronik mit SW-Anteil) (1)
- Von innen über Hebel (Mechanik)
- Von innen über Schalter an Türe oder in Mittelkonsole (Mechatronik mit SW-Anteil) (2)
- Von innen über Airbagsensor (Mechatronik mit SW-Anteil) (3)

■ Funktion "Tür verriegeln"

- Von aussen über Schlüssel (Mechanik)
- Von aussen über Funkschlüssel (Mechatronik mit SW-Anteil) (1)
- Von innen über Schalter an Türe oder in Mittelkonsole (Mechatronik mit SW-Anteil) (4)
- Von innen über Knopf (Mechanik)
- Von innen zeitgesteuert (Mechatronik mit SW-Anteil) (5)
- Von innen geschwindigkeitsgesteuert (Mechatronik mit SW-Anteil) (6)

■ Komfortfunktionen

- Einzeltür / Zentral für alle Türen
- Kindersicherung

- Ein- und Ausschalten der Geschwindigkeitssteuerung (SW) (7)

- Anzeige des Verriegelungszustands im Display (SW) (8)

- Einklemmschutz (Fenster) (Mechatronik mit SW-Anteil)

- Realisierung der zusätzlichen Funktionen über Mechatronik mit SW-Anteil

- Technisch machbar und zumindest prototypisch realisiert

- Öffnen und schliessen der Tür (bei Heckklappe in der Oberklasse Standard)

- Entriegeln, verriegeln und abfragen des Verriegelungszustands über

- GSM / SMS

- Internet und GSM, UMTS, WLAN, ...

- Jeweils ereignis, zeit- oder ortsgesteuert

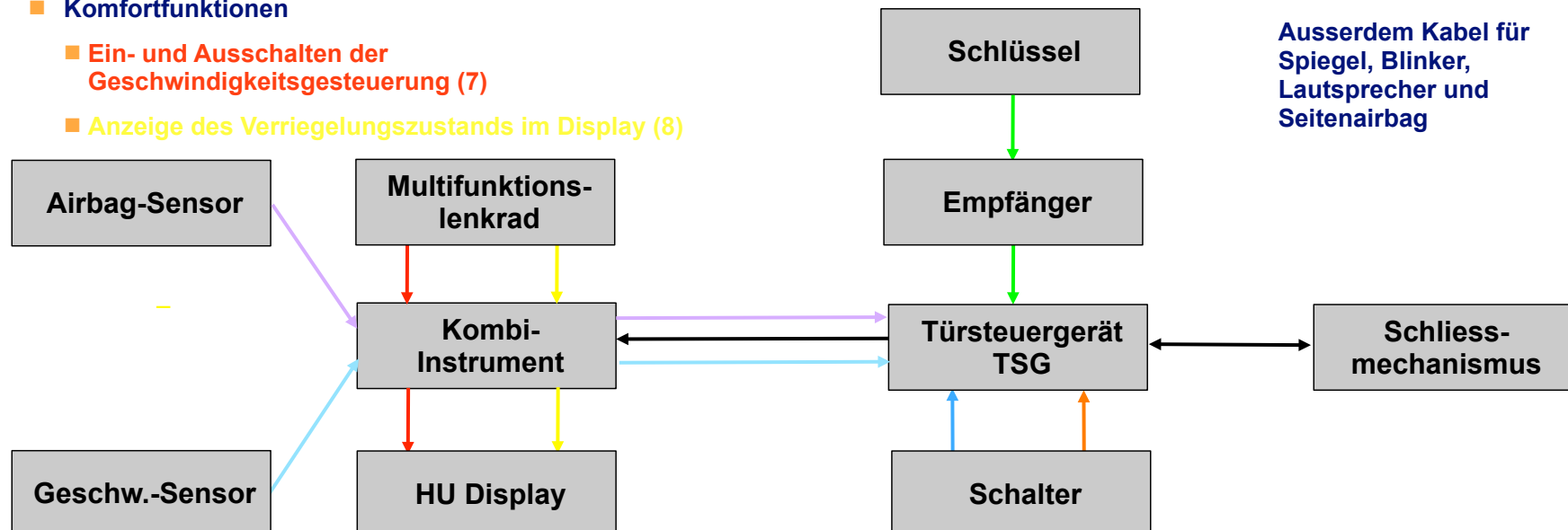
- Aussenspiegel verstellbar und heizbar, Blinker integriert

- Seitenairbag, Lautsprecher

Türsteuerung: Steuergeräte, Aktoren, Sensoren Struktur und Vernetzung



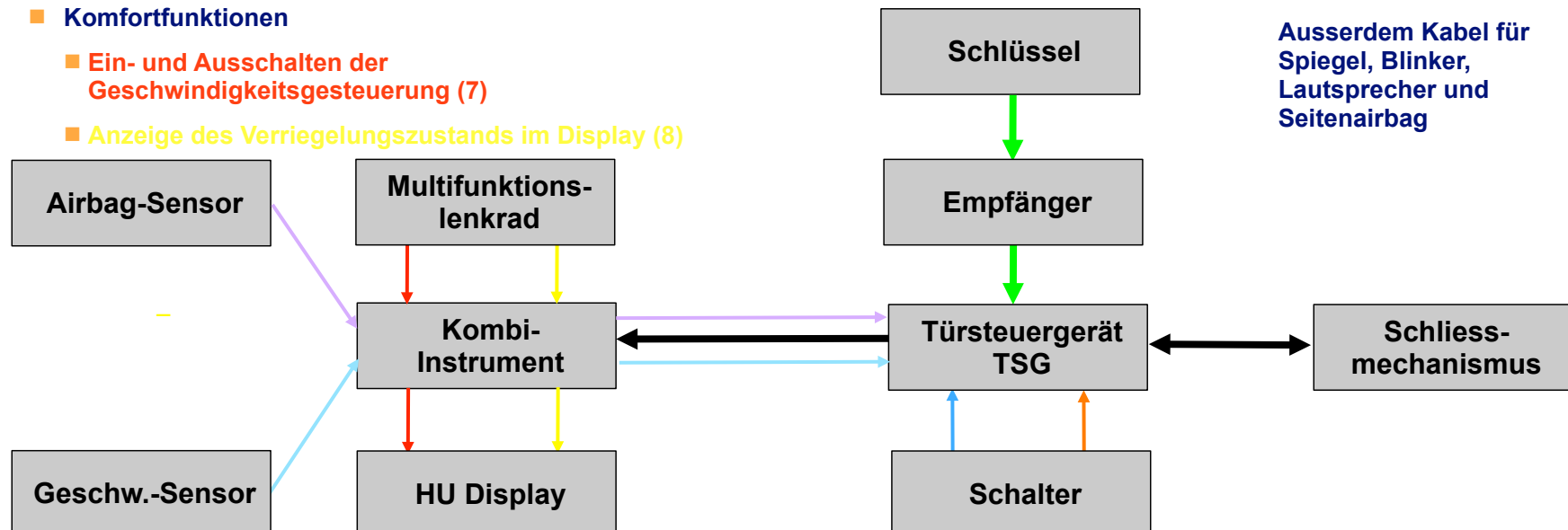
- Funktion “Tür entriegeln”
 - Von aussen über Funkschlüssel (1)
 - Von innen über Schalter (2)
 - Von innen über Airbagsensor (3)
- Funktion “Tür verriegeln”
 - Von aussen über Funkschlüssel (1)
 - Von innen über Schalter (4)
 - Von innen zeitgesteuert (5)
 - Von innen geschwindigkeitsgesteuert (6)
- Komfortfunktionen
 - Ein- und Ausschalten der Geschwindigkeitssteuerung (7)
 - Anzeige des Verriegelungszustands im Display (8)



Türsteuerung: Steuergeräte, Aktoren, Sensoren Struktur und Vernetzung



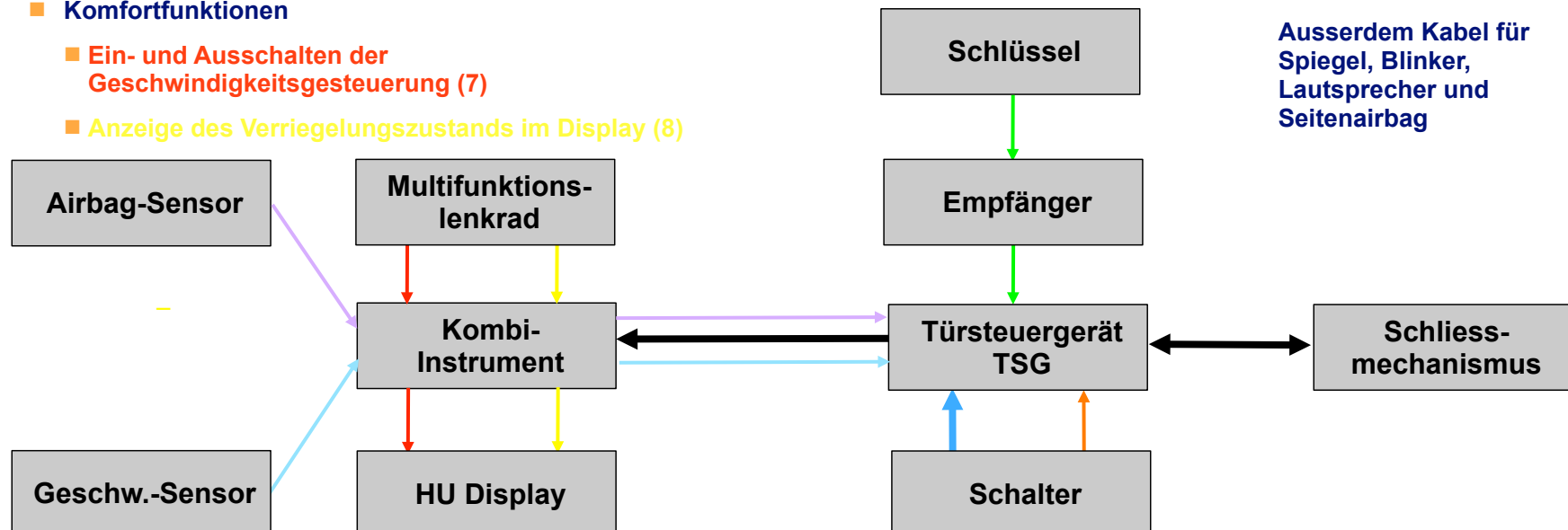
- Funktion “Tür entriegeln”
 - Von aussen über Funkschlüssel (1)
 - Von innen über Schalter (2)
 - Von innen über Airbagsensor (3)
- Funktion “Tür verriegeln”
 - Von aussen über Funkschlüssel (1)
 - Von innen über Schalter (4)
 - Von innen zeitgesteuert (5)
 - Von innen geschwindigkeitsgesteuert (6)
- Komfortfunktionen
 - Ein- und Ausschalten der Geschwindigkeitssteuerung (7)
 - Anzeige des Verriegelungszustands im Display (8)



Türsteuerung: Steuergeräte, Aktoren, Sensoren Struktur und Vernetzung



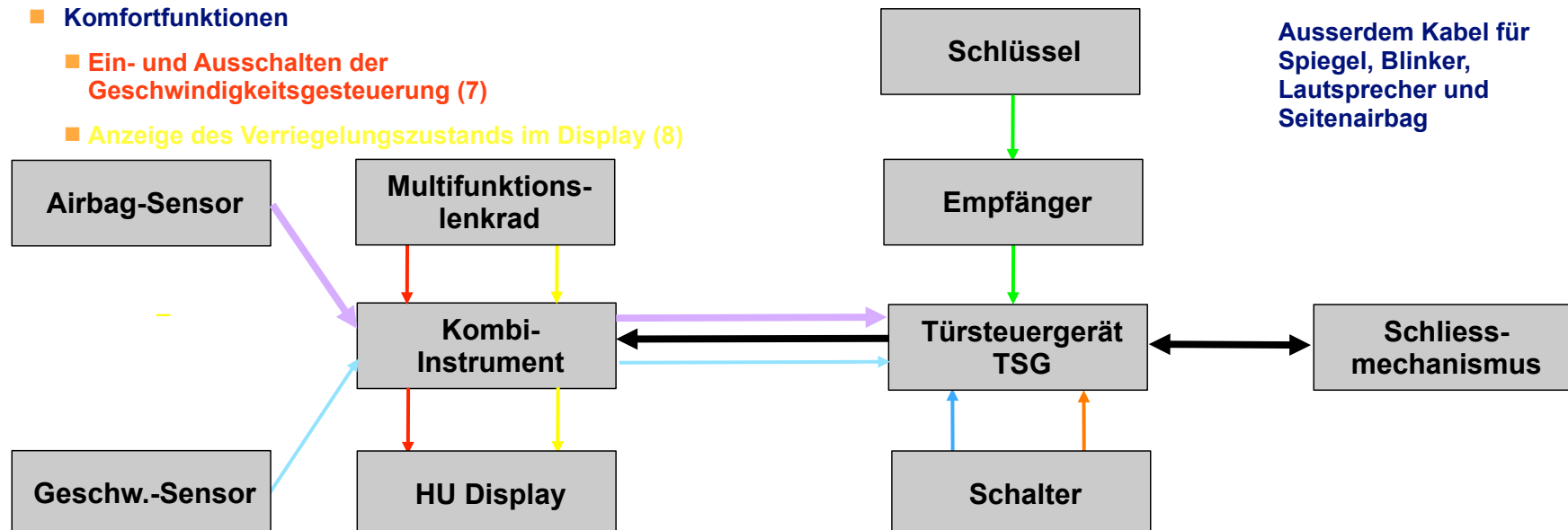
- Funktion “Tür entriegeln”
 - Von aussen über Funkschlüssel (1)
 - Von innen über Schalter (2)
 - Von innen über Airbagsensor (3)
- Funktion “Tür verriegeln”
 - Von aussen über Funkschlüssel (1)
 - Von innen über Schalter (4)
 - Von innen zeitgesteuert (5)
 - Von innen geschwindigkeitsgesteuert (6)
- Komfortfunktionen
 - Ein- und Ausschalten der Geschwindigkeitssteuerung (7)
 - Anzeige des Verriegelungszustands im Display (8)



Türsteuerung: Steuergeräte, Aktoren, Sensoren Struktur und Vernetzung



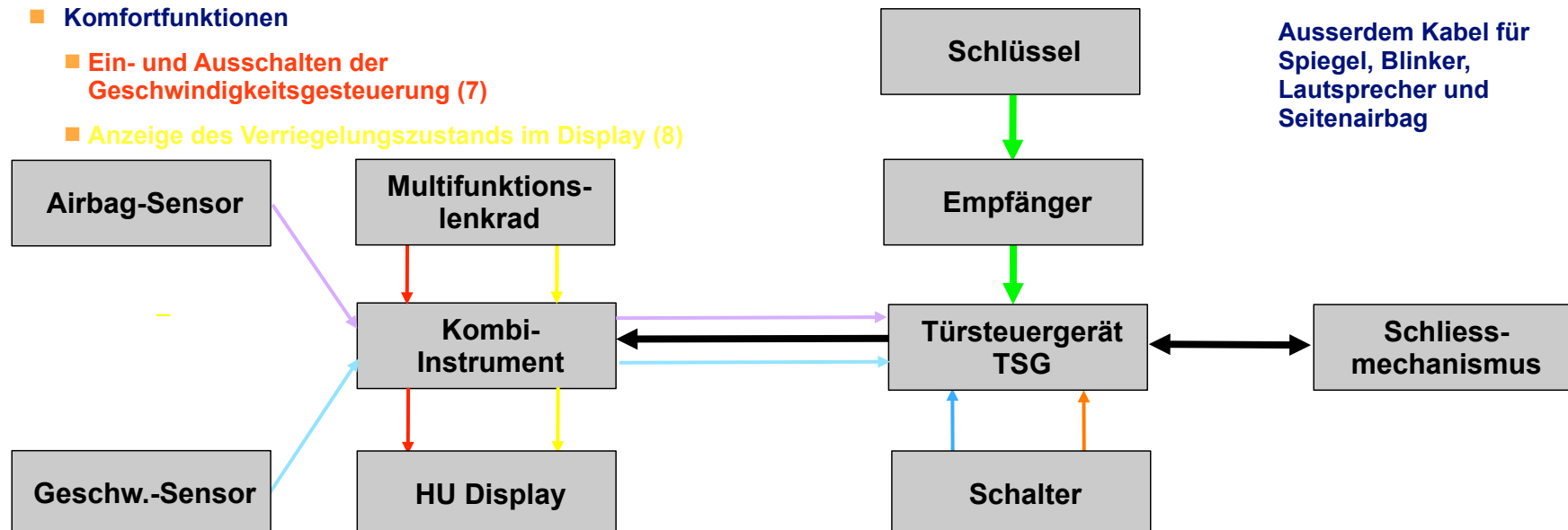
- Funktion “Tür entriegeln”
 - Von aussen über Funkschlüssel (1)
 - Von innen über Schalter (2)
 - Von innen über Airbagsensor (3)
- Funktion “Tür verriegeln”
 - Von aussen über Funkschlüssel (1)
 - Von innen über Schalter (4)
 - Von innen zeitgesteuert (5)
 - Von innen geschwindigkeitsgesteuert (6)
- Komfortfunktionen
 - Ein- und Ausschalten der Geschwindigkeitssteuerung (7)
 - Anzeige des Verriegelungszustands im Display (8)



Türsteuerung: Steuergeräte, Aktoren, Sensoren Struktur und Vernetzung



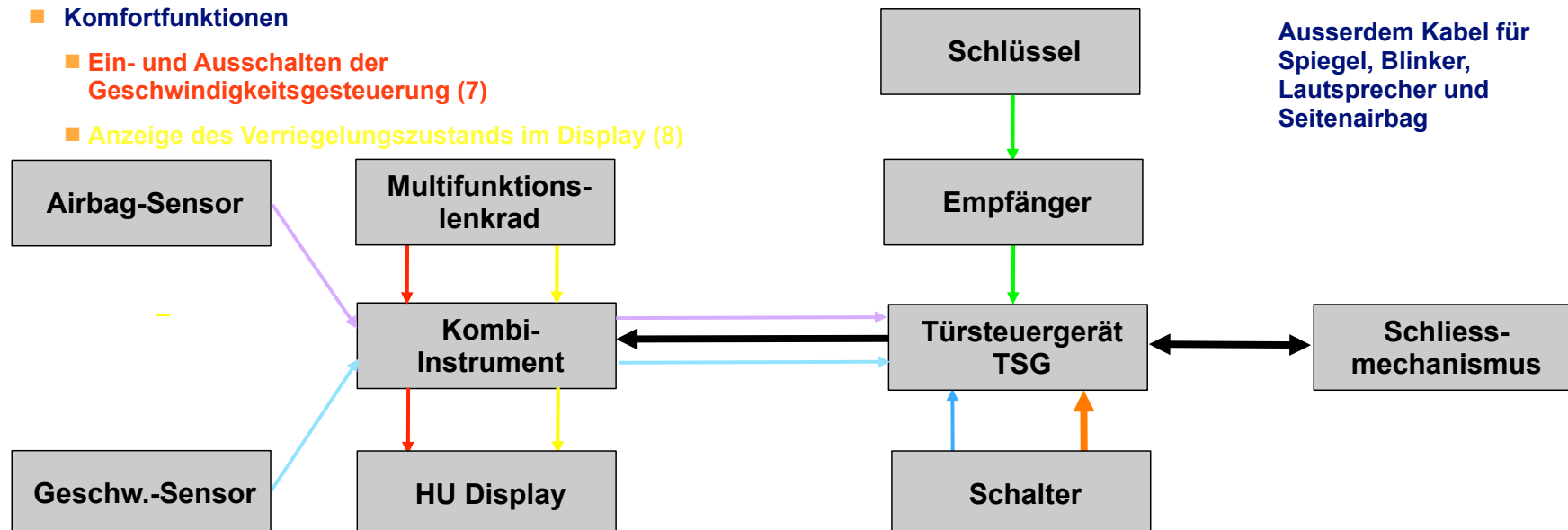
- Funktion “Tür entriegeln”
 - Von aussen über Funkschlüssel (1)
 - Von innen über Schalter (2)
 - Von innen über Airbagsensor (3)
- Funktion “Tür verriegeln”
 - Von aussen über Funkschlüssel (1)
 - Von innen über Schalter (4)
 - Von innen zeitgesteuert (5)
 - Von innen geschwindigkeitsgesteuert (6)
- Komfortfunktionen
 - Ein- und Ausschalten der Geschwindigkeitssteuerung (7)
 - Anzeige des Verriegelungszustands im Display (8)



Türsteuerung: Steuergeräte, Aktoren, Sensoren Struktur und Vernetzung



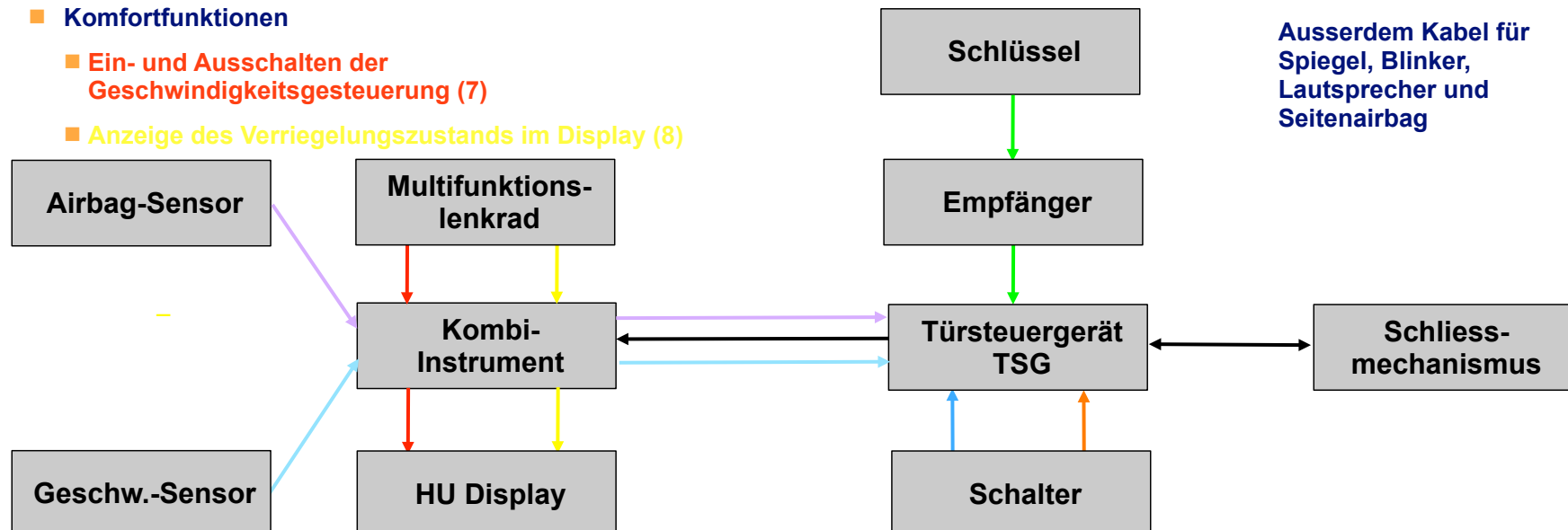
- Funktion “Tür entriegeln”
 - Von aussen über Funkschlüssel (1)
 - Von innen über Schalter (2)
 - Von innen über Airbagsensor (3)
- Funktion “Tür verriegeln”
 - Von aussen über Funkschlüssel (1)
 - Von innen über Schalter (4)
 - Von innen zeitgesteuert (5)
 - Von innen geschwindigkeitsgesteuert (6)
- Komfortfunktionen
 - Ein- und Ausschalten der Geschwindigkeitssteuerung (7)
 - Anzeige des Verriegelungszustands im Display (8)



Türsteuerung: Steuergeräte, Aktoren, Sensoren Struktur und Vernetzung



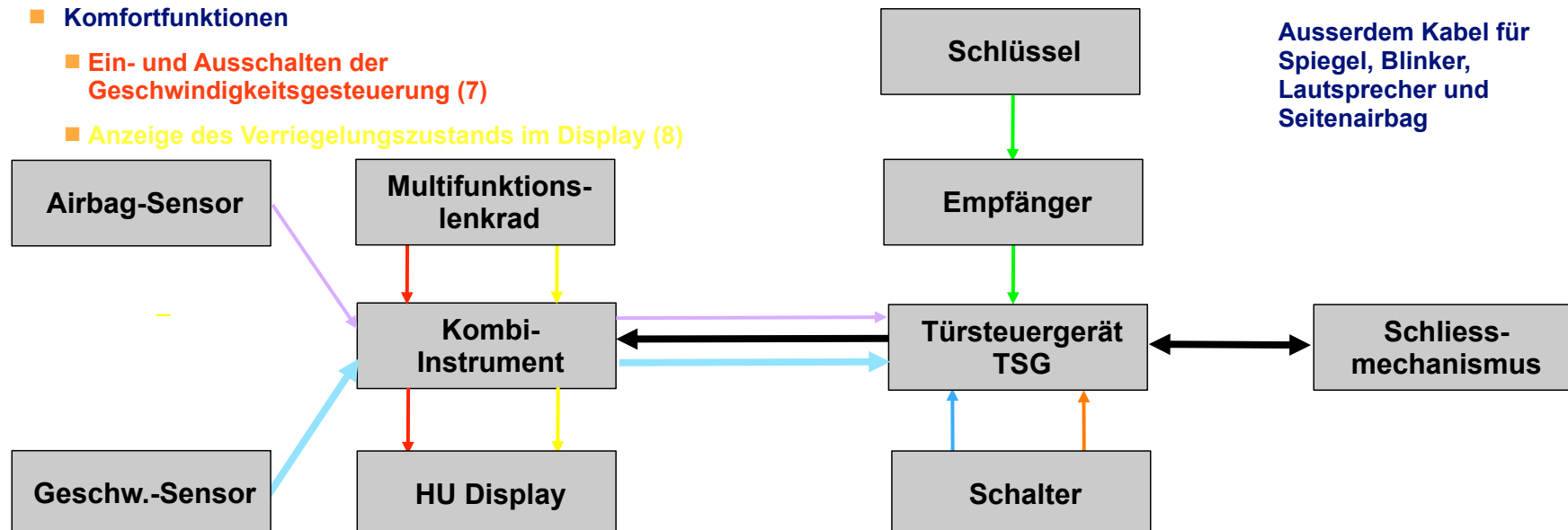
- Funktion “Tür entriegeln”
 - Von aussen über Funkschlüssel (1)
 - Von innen über Schalter (2)
 - Von innen über Airbagsensor (3)
- Funktion “Tür verriegeln”
 - Von aussen über Funkschlüssel (1)
 - Von innen über Schalter (4)
 - Von innen zeitgesteuert (5)
 - Von innen geschwindigkeitsgesteuert (6)
- Komfortfunktionen
 - Ein- und Ausschalten der Geschwindigkeitssteuerung (7)
 - Anzeige des Verriegelungszustands im Display (8)



Türsteuerung: Steuergeräte, Aktoren, Sensoren Struktur und Vernetzung



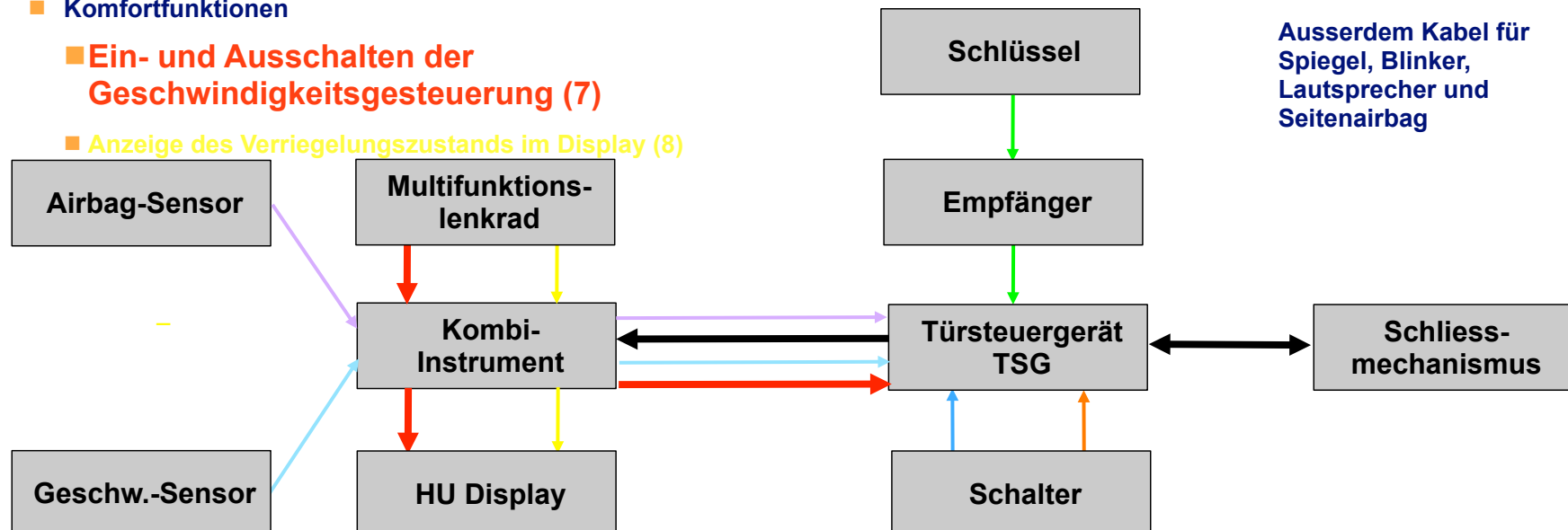
- Funktion “Tür entriegeln”
 - Von aussen über Funkschlüssel (1)
 - Von innen über Schalter (2)
 - Von innen über Airbagsensor (3)
- Funktion “Tür verriegeln”
 - Von aussen über Funkschlüssel (1)
 - Von innen über Schalter (4)
 - Von innen zeitgesteuert (5)
 - Von innen geschwindigkeitsgesteuert (6)
- Komfortfunktionen
 - Ein- und Ausschalten der Geschwindigkeitssteuerung (7)
 - Anzeige des Verriegelungszustands im Display (8)



Türsteuerung: Steuergeräte, Aktoren, Sensoren Struktur und Vernetzung



- Funktion “Tür entriegeln”
 - Von aussen über Funkschlüssel (1)
 - Von innen über Schalter (2)
 - Von innen über Airbagsensor (3)
- Funktion “Tür verriegeln”
 - Von aussen über Funkschlüssel (1)
 - Von innen über Schalter (4)
 - Von innen zeitgesteuert (5)
 - Von innen geschwindigkeitsgesteuert (6)
- Komfortfunktionen
 - Ein- und Ausschalten der Geschwindigkeitssteuerung (7)
 - Anzeige des Verriegelungszustands im Display (8)



Türsteuerung: Steuergeräte, Aktoren, Sensoren Struktur und Vernetzung



■ Funktion "Tür entriegeln"

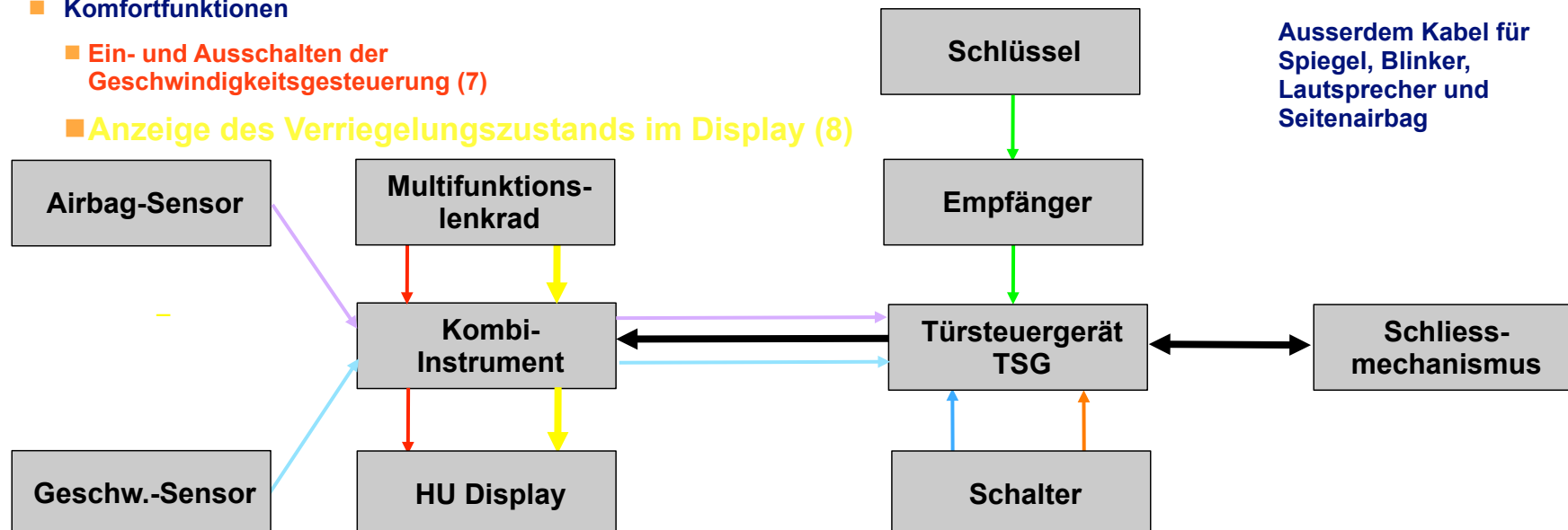
- Von aussen über Funkschlüssel (1)
- Von innen über Schalter (2)
- Von innen über Airbagsensor (3)

■ Funktion "Tür verriegeln"

- Von aussen über Funkschlüssel (1)
- Von innen über Schalter (4)
- Von innen zeitgesteuert (5)
- Von innen geschwindigkeitsgesteuert (6)

■ Komfortfunktionen

- Ein- und Ausschalten der Geschwindigkeitssteuerung (7)
- Anzeige des Verriegelungszustands im Display (8)



Türsteuerung: Standard bis ca. 1990



■ Funktion “Tür entriegeln”

- Von aussen über Schlüssel
- Von innen über Hebel

■ Funktion “Tür verriegeln”

- Von aussen über Schlüssel
- Von innen über Knopf

■ Komfortfunktion

- Einzeltür / Zentral für alle Türen
- Kindersicherung

■ Realisierung der Funktionen über Mechanik

■ Kein SW-Anteil

Türsteuergerät (TSG)



- Steuert (auch) die Türe
 - Verriegeln
 - Entriegeln
- Steuert daneben meistens
 - Fensterheber
 - Einklemmschutz
 - Aussenspiegel
- In manchen Modellen: Ansteuerung von
 - Sitzverstellung
 - Sitzheizung
- Eigentlich „Steuergerät in der Türe“ (zur Zeit meistens)

■ Ansprechpartner

- Baugruppenverantwortlicher Karosserie
- Baugruppenverantwortlicher Sitze
- Baugruppenverantwortlicher Kombi-Instrument
- Baugruppenverantwortlicher Blinker
- Baugruppenverantwortlicher Mittelkonsole
- Baugruppenverantwortlicher Soundsystem
- Baugruppenverantwortlicher Seitenairbag
- Verantwortlicher Passive Sicherheit
- Verantwortlicher EMV
- Verantwortlicher Verkabelung
- Verantwortlicher Vernetzung
- Verantwortlicher Telematik

■ Zulieferer

- Schliesssystem
- Scheiben
- Fensterheber
- Aussenspiegel
- Türsteuergerät
- Schalter

■ Schnittstellen

- Mechanik
- Energie
- Information

Baugruppenverantwortlicher Türe (bis ca. 1990)



■ Ansprechpartner

- Baugruppenverantwortlicher Karosserie

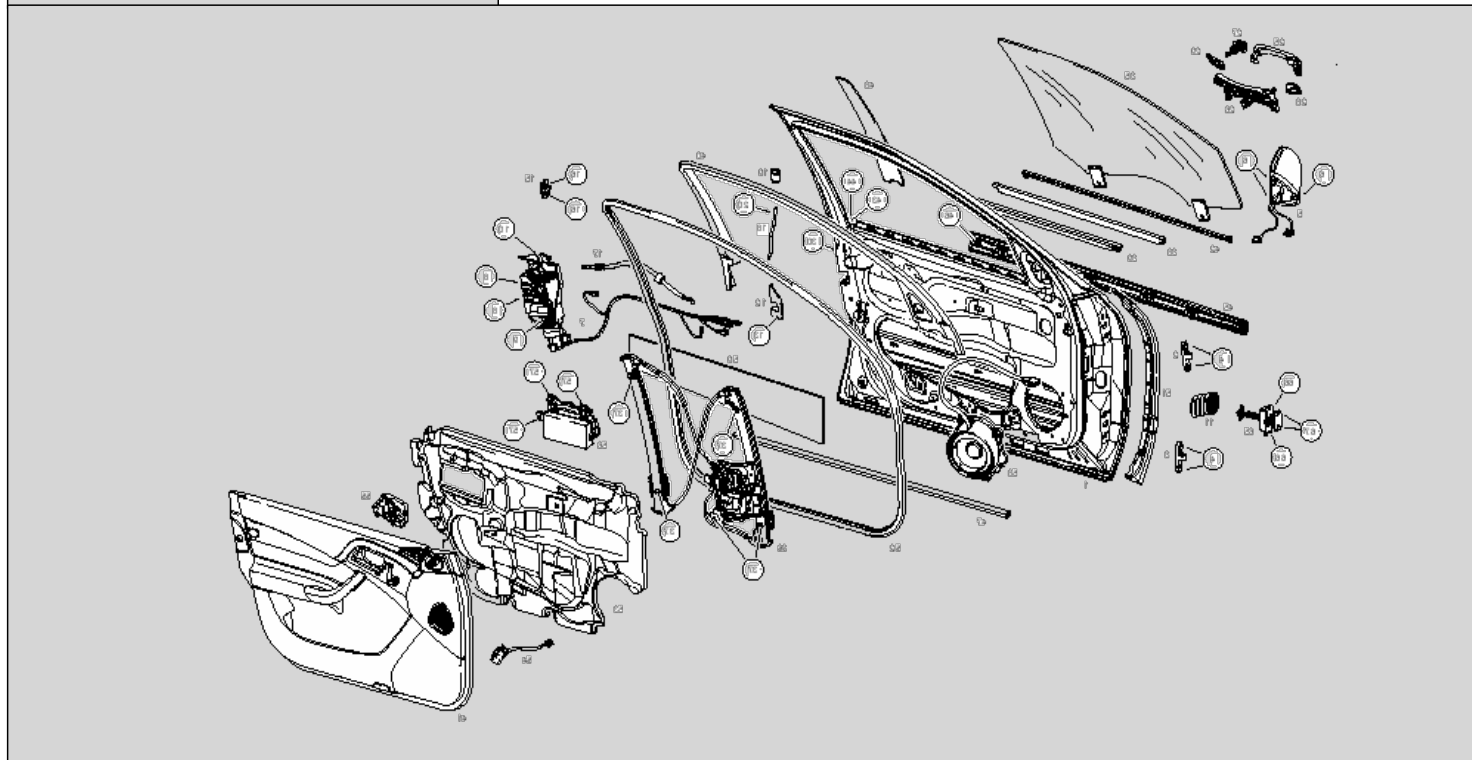
■ Zulieferer

- Schliesssystem
- Scheiben
- Fensterheber
- Aussenspiegel

■ Schnittstellen

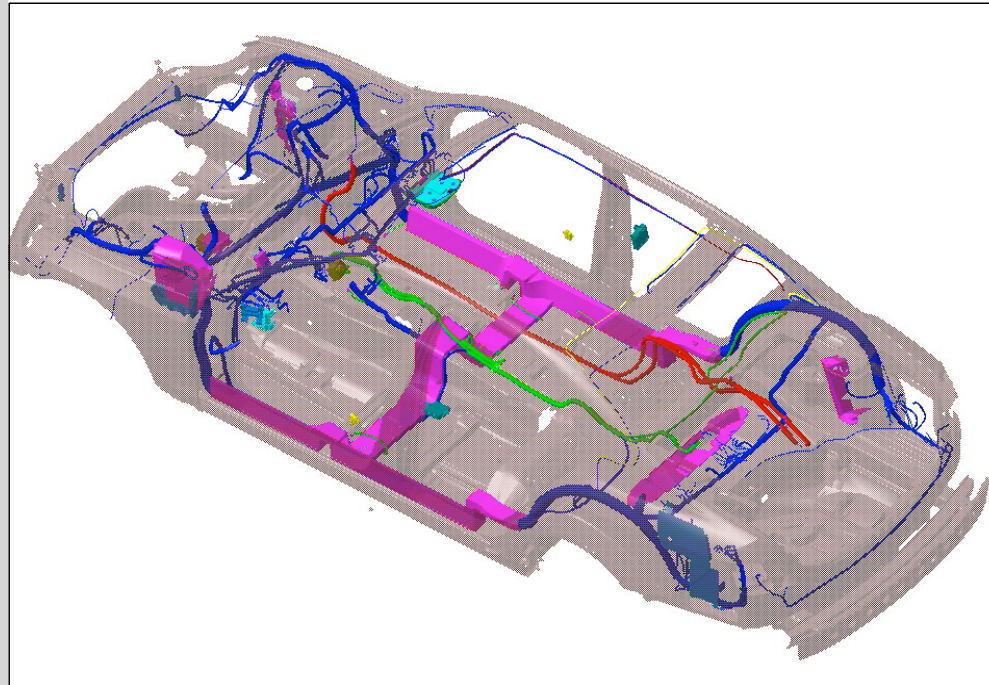
- Mechanik

Packaging/Montagemodule



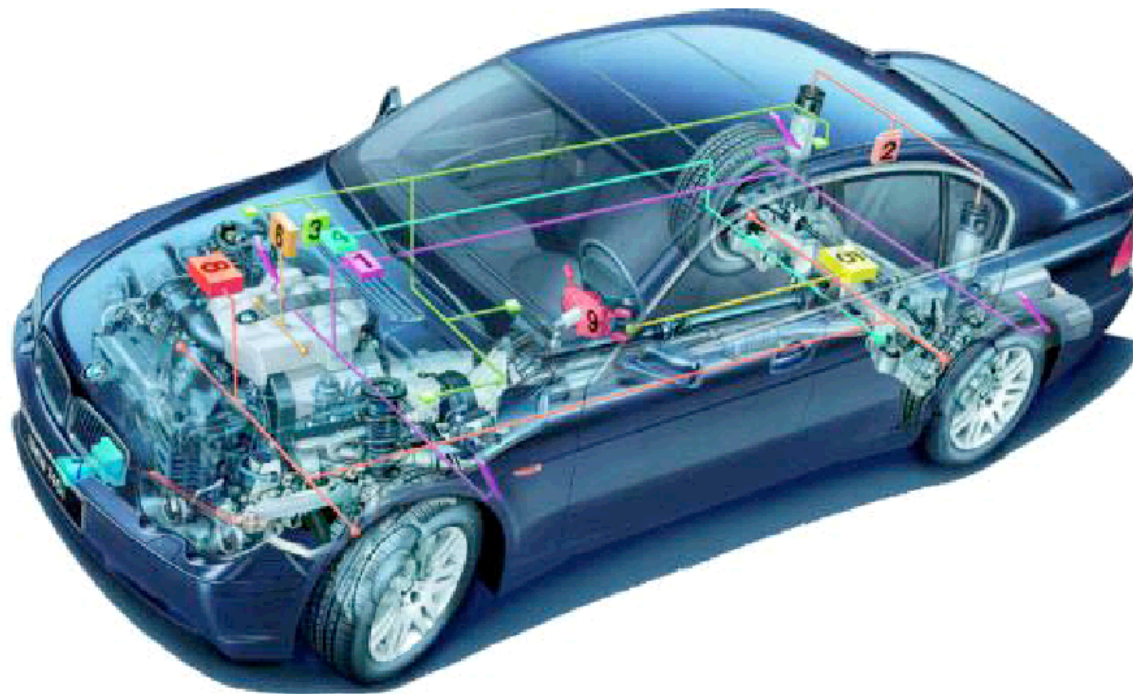
Packaging/Montagemodule

Verkabelung



Der Kabelbaum eines modernen Fahrzeugs besteht aus drei funktionalen Gruppen:

1. Elektrische Energieversorgung (Energiebordnetz)
2. Informationstechnische Verbindung zwischen den Systemen (Bussysteme)
3. HF-Verbindungen von den Antennen zu den Endgeräten



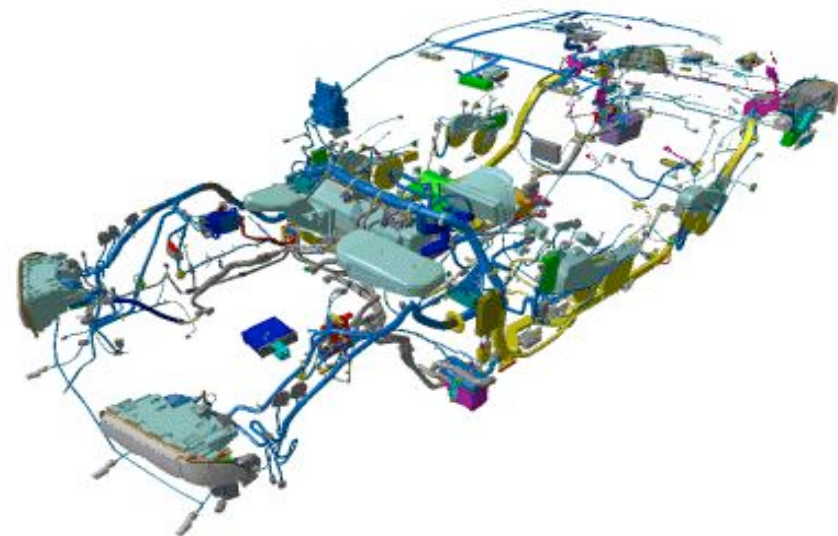
Die Topologie ergibt sich aus Optimierungszielen:

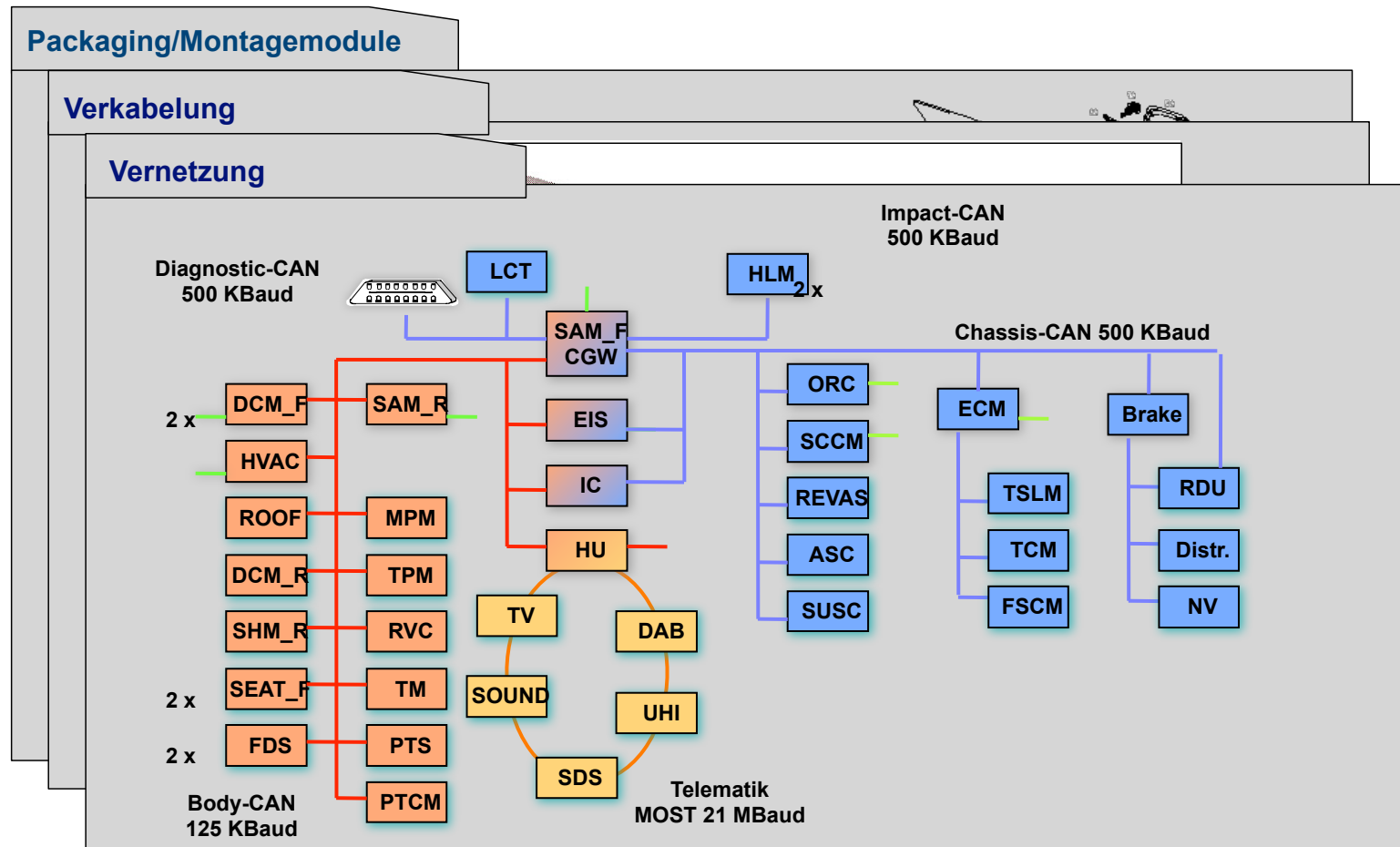
1. Kosten
2. Gewicht
3. Montagefreundlichkeit
4. Betriebssicherheit

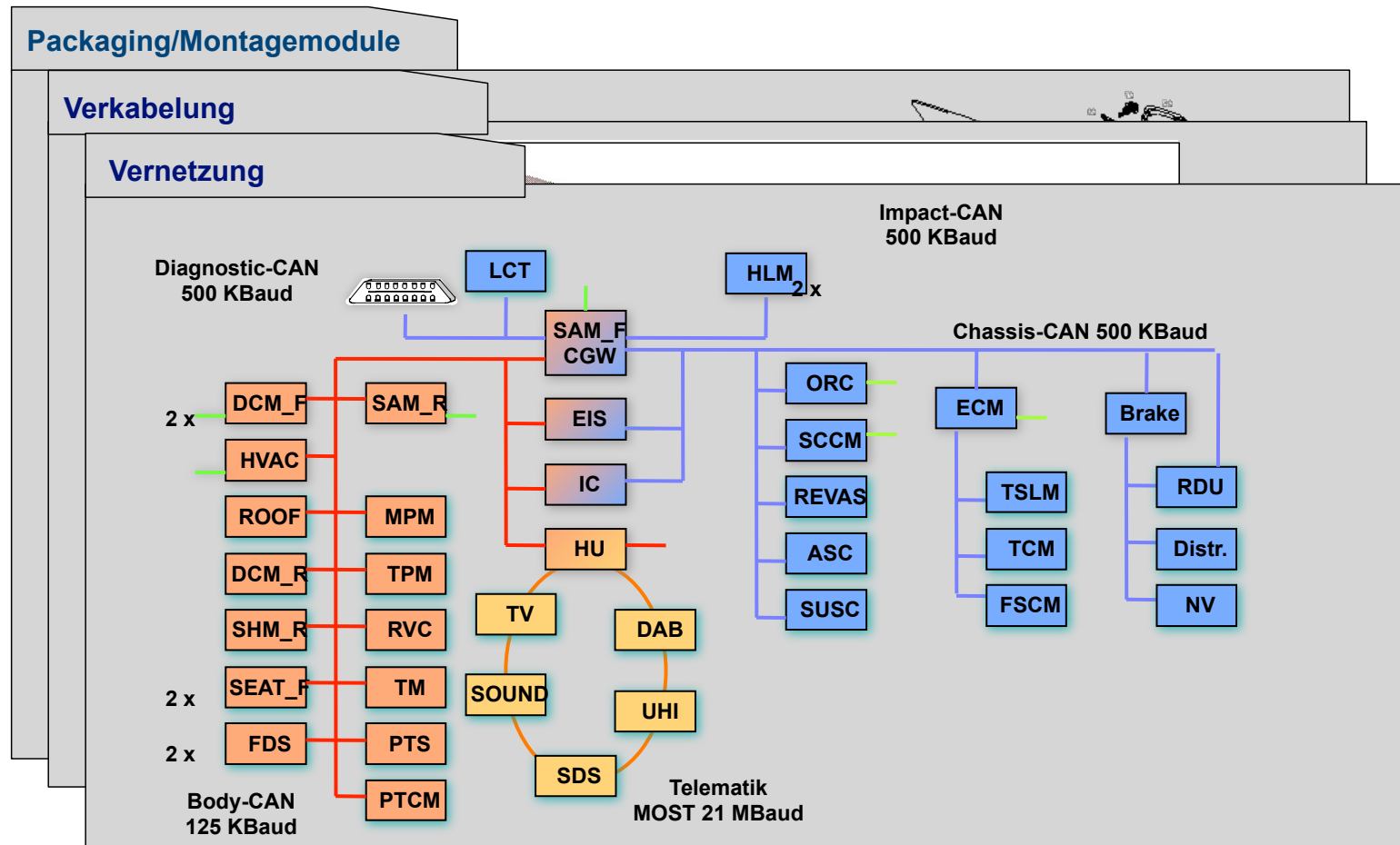
Das physikalische Bordnetz zählt zu den aufwändigsten, teuersten und schwersten Komponenten in modernen KFZ.

Beispiel BMW 5er Modelljahr 2003

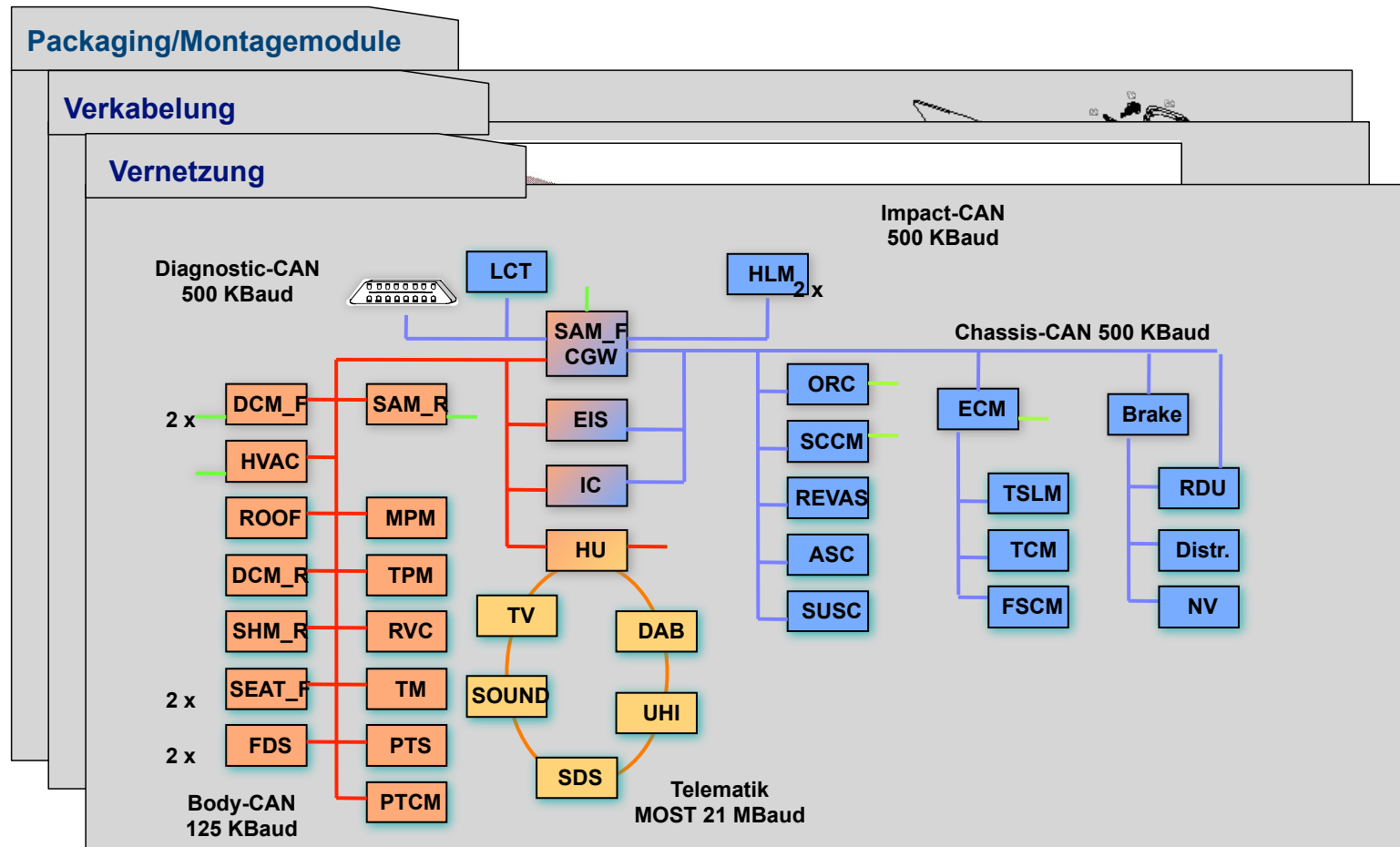
- Länge 7,3 km
- Masse 55 kg

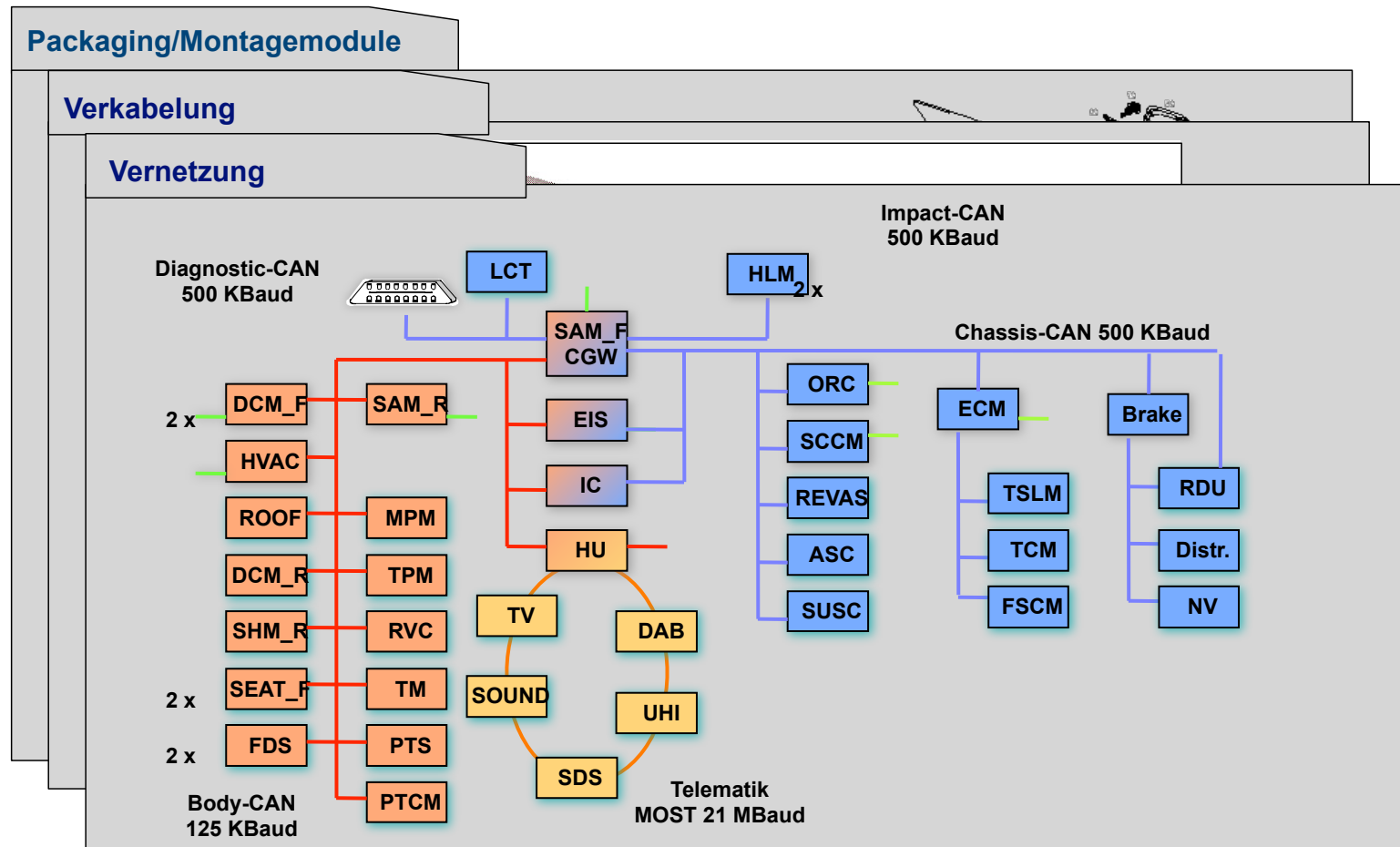






CAN





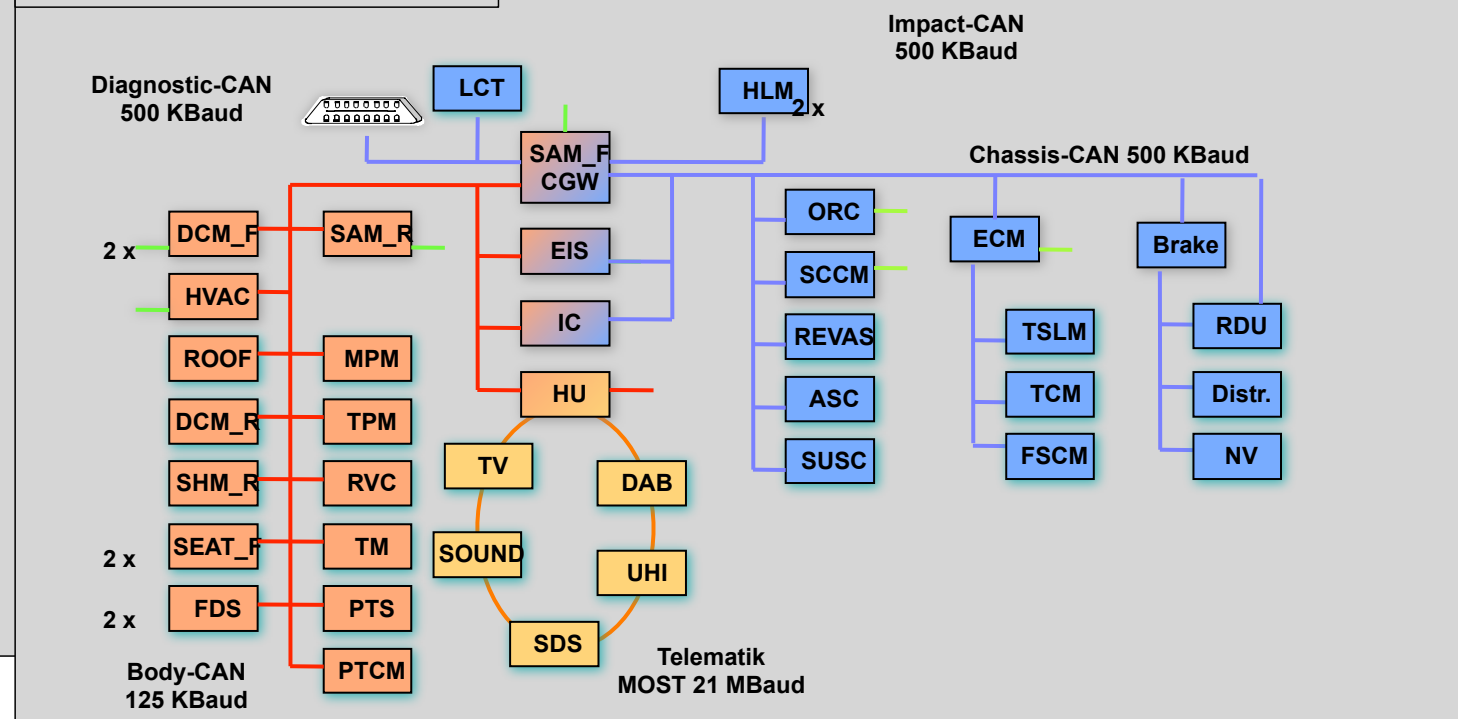
Komplexitätsproblem



Packaging/Montagemodule

Verkabelung

Vernetzung

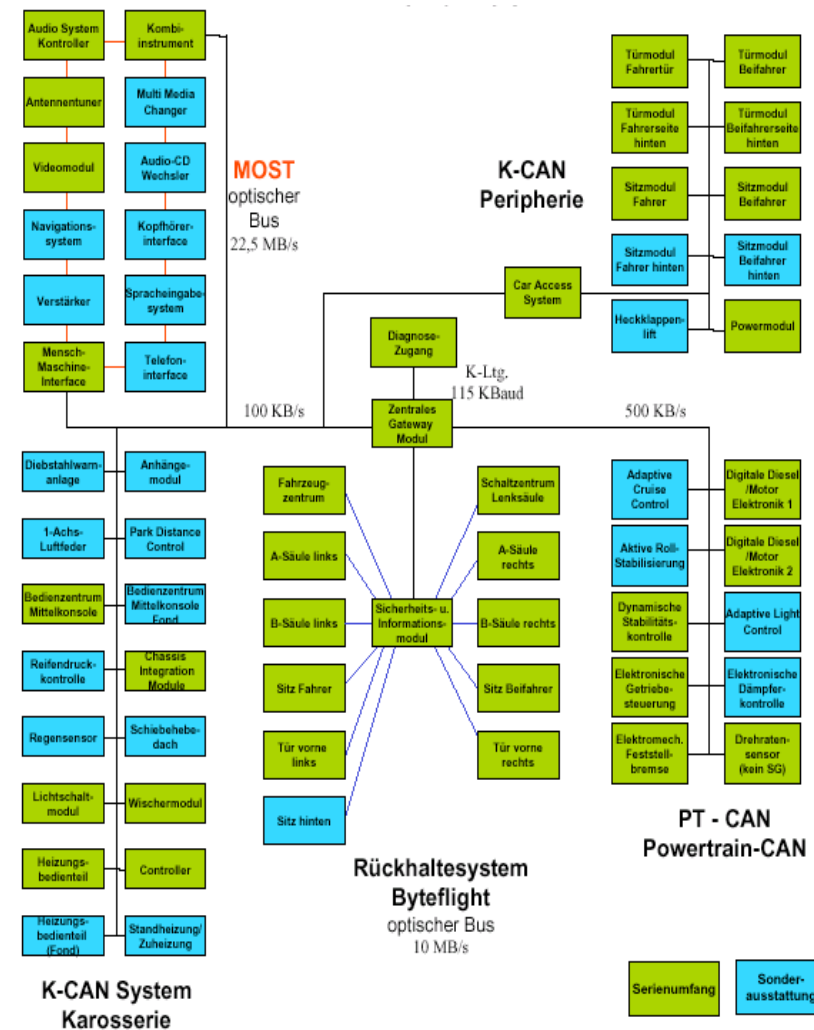


Protokolle und Bussysteme: Beispiele



BMW 7er, Modelljahr 2001 (E65)

- Powertrain: Highspeed-CAN
- Karosserie und Peripherie: Lowspeed-CAN
- Infotainment: MOST
- Passive Sicherheit: byteflight
- Motor: Highspeed-CAN (nicht gezeigt)
- Diagnose: K-Line
- Backups: K-Line
 - Airbag-Telefon, Blinkerhebel-LSZ, Gangwahl-EGS, DSC-ABS,...)
- Gateways
 - ZGM (byteflight, K-CAN, PT-CAN, Diagnose)
 - DME (PT-CAN, LoCAN)
 - MMI (K-CAN, MOST)
 - Kombi (K-CAN, MOST)

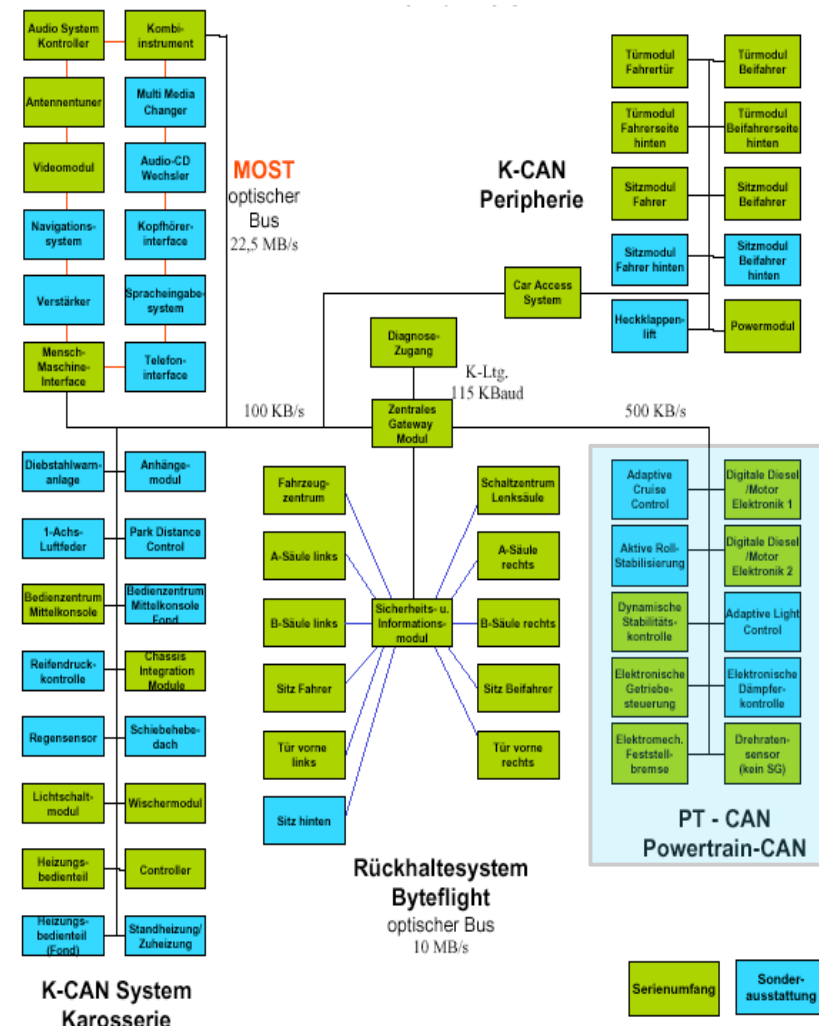


Protokolle und Bussysteme: Beispiele



BMW 7er, Modelljahr 2001 (E65)

- **Powertrain: Highspeed-CAN**
- Karosserie und Peripherie: Lowspeed-CAN
- Infotainment: MOST
- Passive Sicherheit: byteflight
- Motor: Highspeed-CAN (nicht gezeigt)
- Diagnose: K-Line
- Backups: K-Line
 - Airbag-Telefon, Blinkerhebel-LSZ, Gangwahl-EGS, DSC-ABS,...)
- Gateways
 - ZGM (byteflight, K-CAN, PT-CAN, Diagnose)
 - DME (PT-CAN, LoCAN)
 - MMI (K-CAN, MOST)
 - Kombi (K-CAN, MOST)

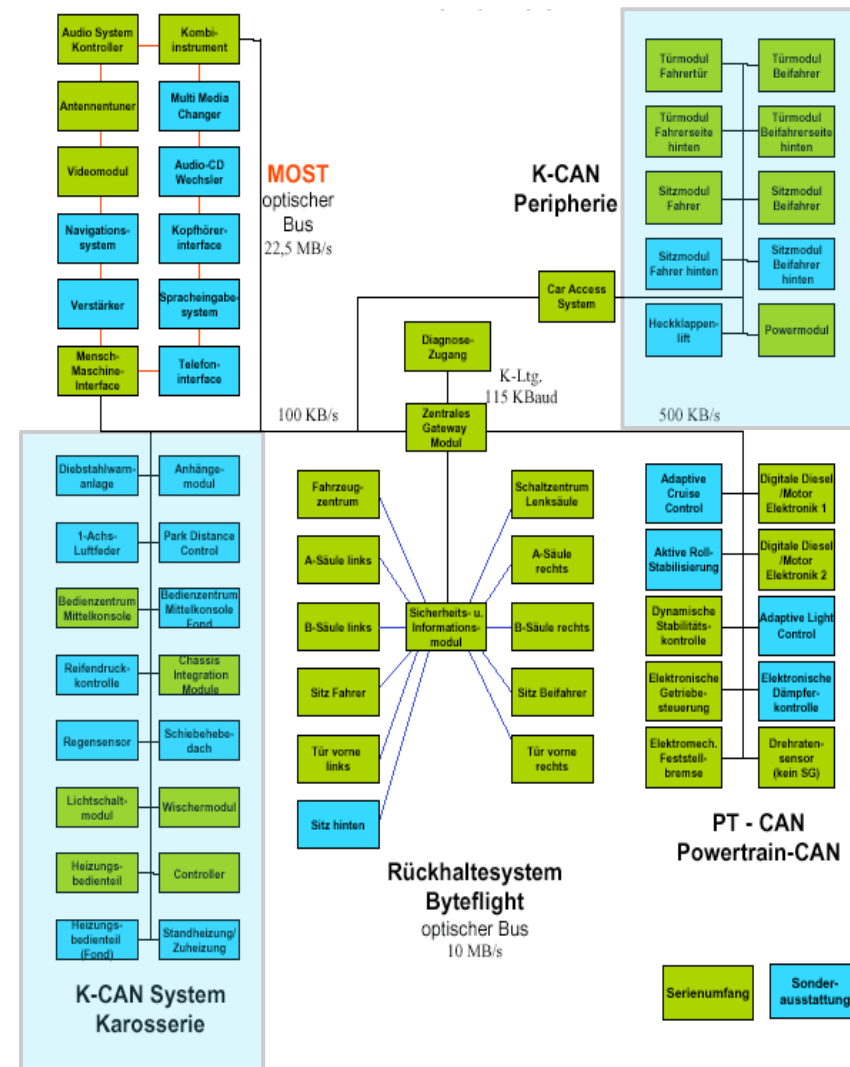


Protokolle und Bussysteme: Beispiele



BMW 7er, Modelljahr 2001 (E65)

- Powertrain: Highspeed-CAN
- **Karosserie und Peripherie: Lowspeed-CAN**
- Infotainment: MOST
- Passive Sicherheit: byteflight
- Motor: Highspeed-CAN (nicht gezeigt)
- Diagnose: K-Line
- Backups: K-Line
 - Airbag-Telefon, Blinkerhebel-LSZ, Gangwahl-EGS, DSC-ABS,...)
- Gateways
 - ZGM (byteflight, K-CAN, PT-CAN, Diagnose)
 - DME (PT-CAN, LoCAN)
 - MMI (K-CAN, MOST)
 - Kombi (K-CAN, MOST)

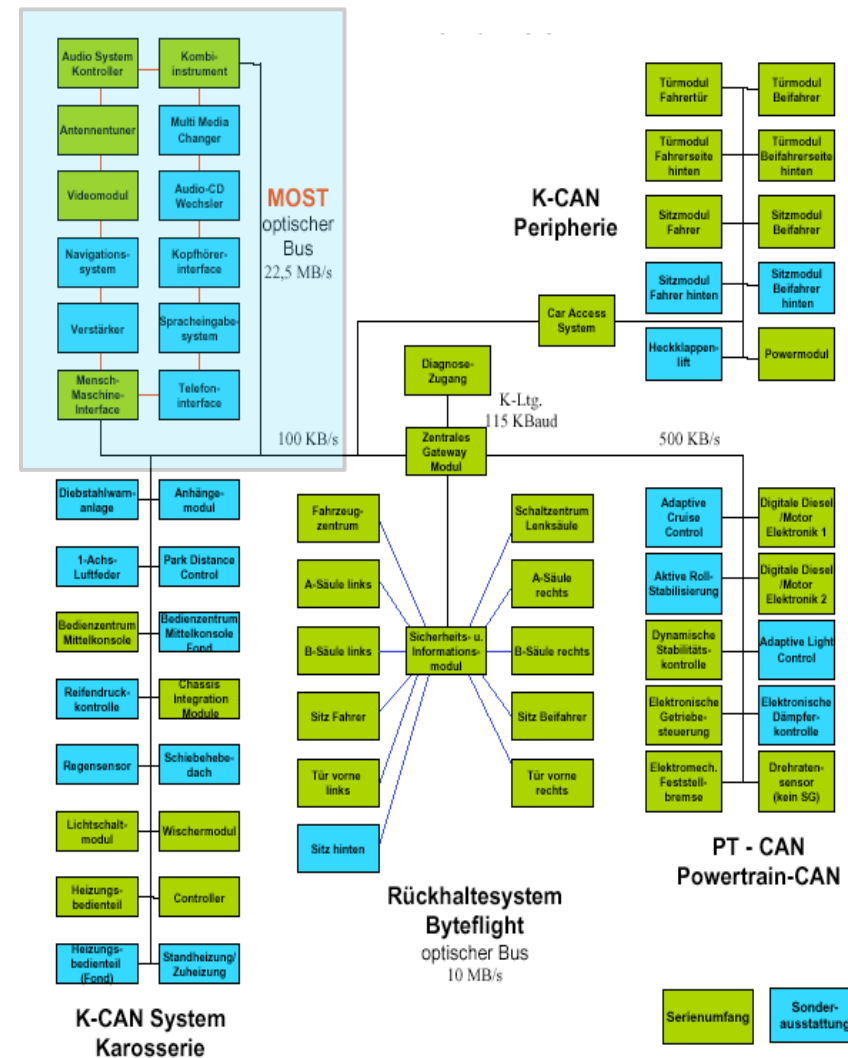


Protokolle und Bussysteme: Beispiele



BMW 7er, Modelljahr 2001 (E65)

- Powertrain: Highspeed-CAN
- Karosserie und Peripherie: Lowspeed-CAN
- Infotainment: MOST
- Passive Sicherheit: byteflight
- Motor: Highspeed-CAN (nicht gezeigt)
- Diagnose: K-Line
- Backups: K-Line
 - Airbag-Telefon, Blinkerhebel-LSZ, Gangwahl-EGS, DSC-ABS,...)
- Gateways
 - ZGM (byteflight, K-CAN, PT-CAN, Diagnose)
 - DME (PT-CAN, LoCAN)
 - MMI (K-CAN, MOST)
 - Kombi (K-CAN, MOST)

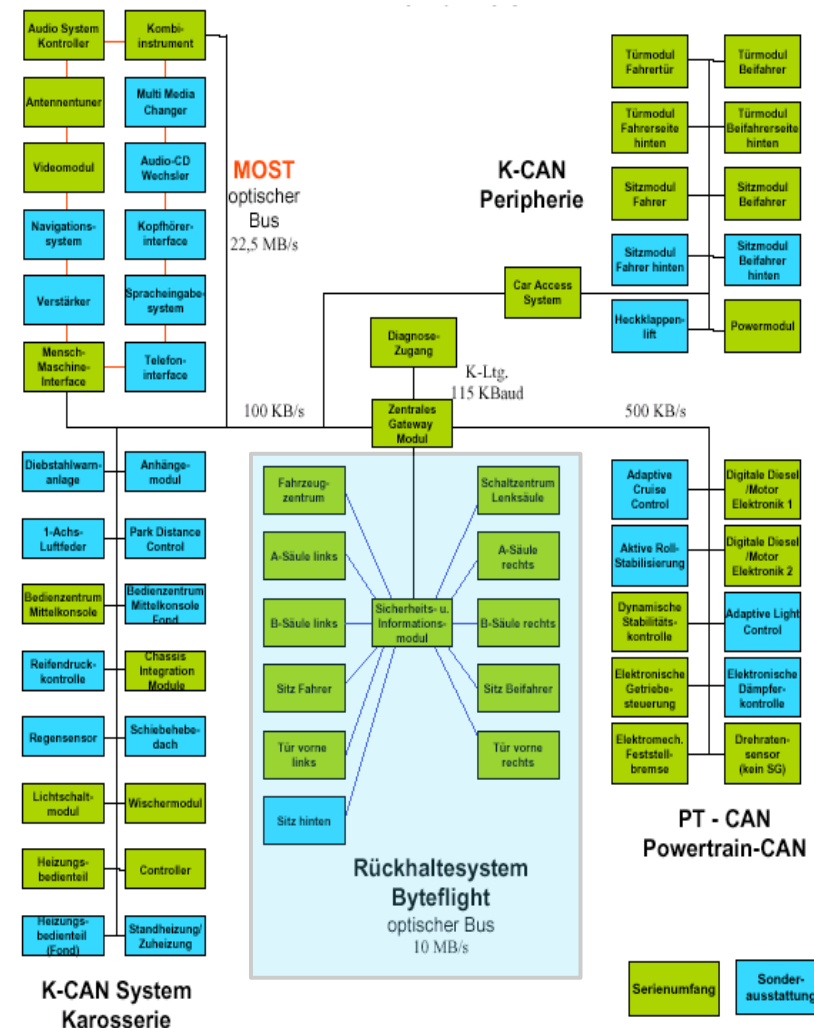


Protokolle und Bussysteme: Beispiele



BMW 7er, Modelljahr 2001 (E65)

- Powertrain: Highspeed-CAN
- Karosserie und Peripherie: Lowspeed-CAN
- Infotainment: MOST
- **Passive Sicherheit: byteflight**
- Motor: Highspeed-CAN (nicht gezeigt)
- Diagnose: K-Line
- Backups: K-Line
 - Airbag-Telefon, Blinkerhebel-LSZ, Gangwahl-EGS, DSC-ABS,...)
- Gateways
 - ZGM (byteflight, K-CAN, PT-CAN, Diagnose)
 - DME (PT-CAN, LoCAN)
 - MMI (K-CAN, MOST)
 - Kombi (K-CAN, MOST)

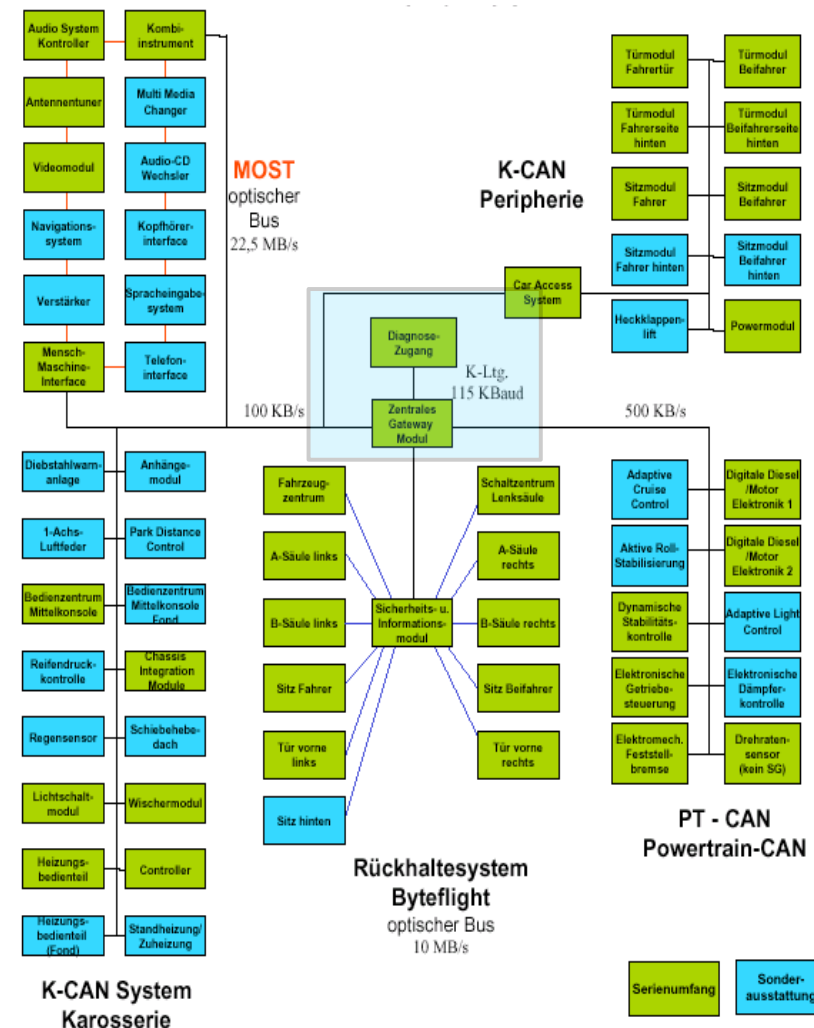


Protokolle und Bussysteme: Beispiele



BMW 7er, Modelljahr 2001 (E65)

- Powertrain: Highspeed-CAN
- Karosserie und Peripherie: Lowspeed-CAN
- Infotainment: MOST
- Passive Sicherheit: byteflight
- Motor: Highspeed-CAN (nicht gezeigt)
- **Diagnose: K-Line**
- Backups: K-Line
 - Airbag-Telefon, Blinkerhebel-LSZ, Gangwahl-EGS, DSC-ABS,...)
- Gateways
 - ZGM (byteflight, K-CAN, PT-CAN, Diagnose)
 - DME (PT-CAN, LoCAN)
 - MMI (K-CAN, MOST)
 - Kombi (K-CAN, MOST)

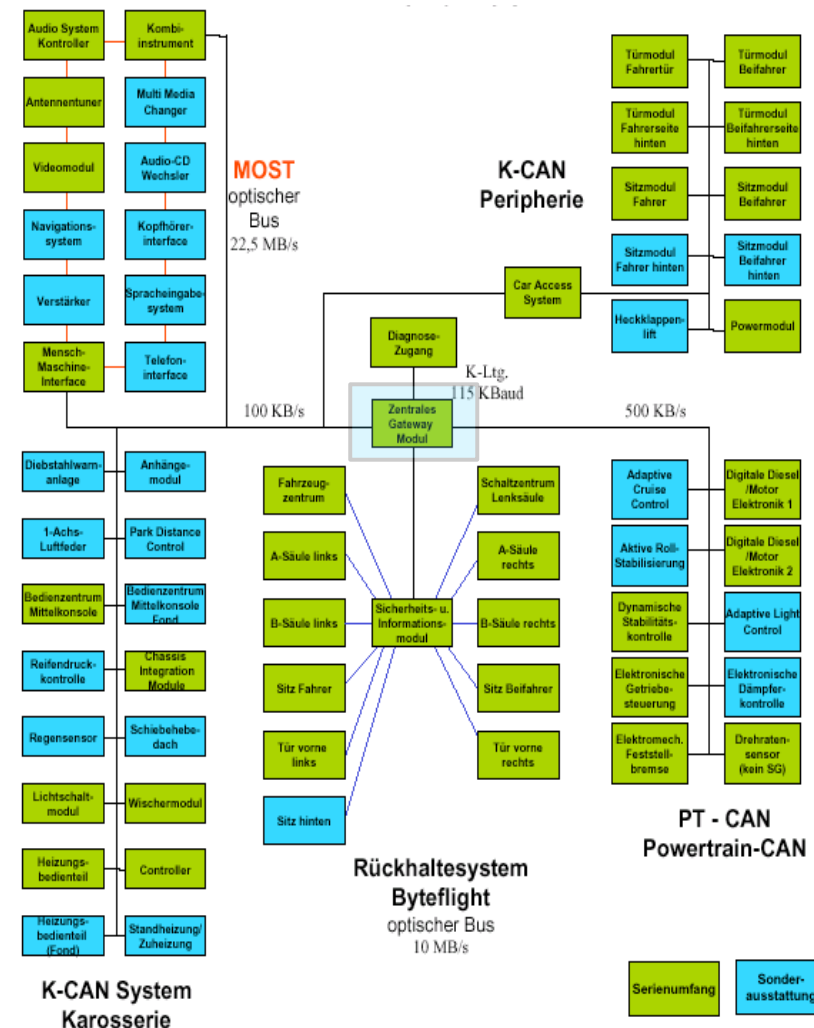


Protokolle und Bussysteme: Beispiele



BMW 7er, Modelljahr 2001 (E65)

- Powertrain: Highspeed-CAN
- Karosserie und Peripherie: Lowspeed-CAN
- Infotainment: MOST
- Passive Sicherheit: byteflight
- Motor: Highspeed-CAN (nicht gezeigt)
- **Diagnose: K-Line**
- Backups: K-Line
 - Airbag-Telefon, Blinkerhebel-LSZ, Gangwahl-EGS, DSC-ABS,...)
- **Gateways**
 - ZGM (byteflight, K-CAN, PT-CAN, Diagnose)
 - DME (PT-CAN, LoCAN)
 - MMI (K-CAN, MOST)
 - Kombi (K-CAN, MOST)



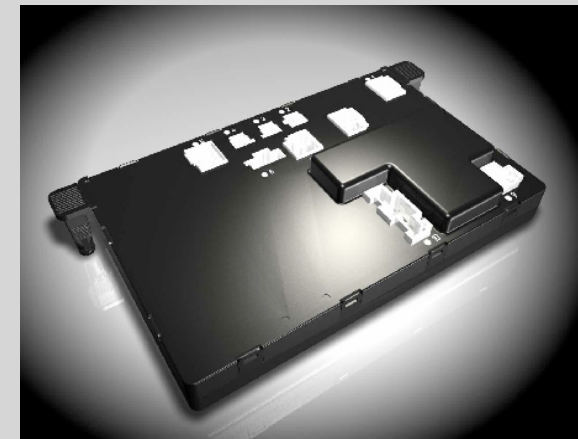
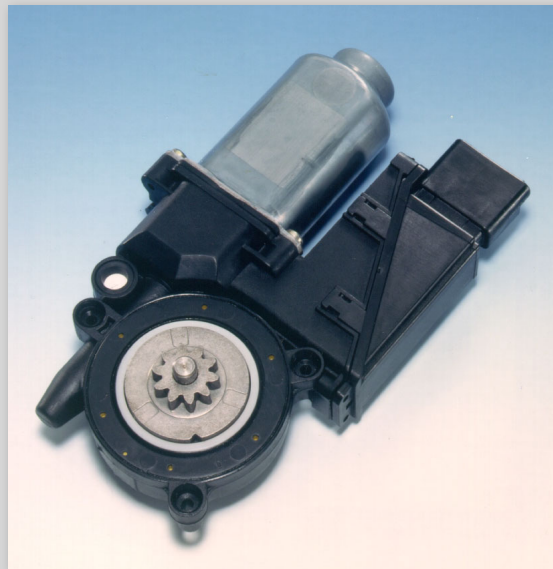
Packaging/Montagemodule

Verkabelung

Vernetzung

Technologien/Plattformen

Impact-CAN
500 Kbaud



Packaging/Montagemodule

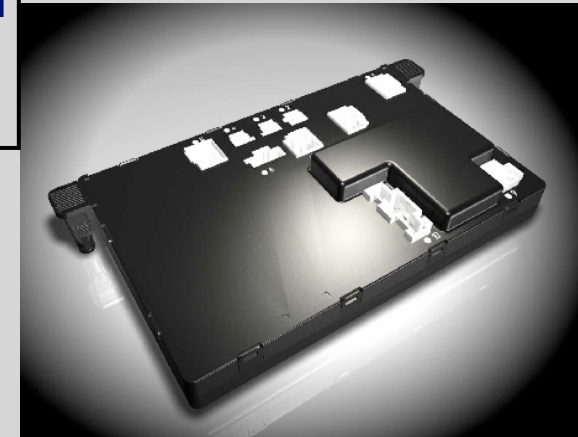
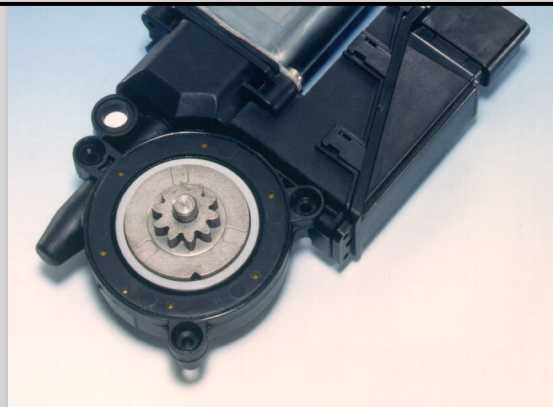
Verkabelung

Vernetzung

Technologien/Plattformen

Impact-CAN
500 Kbaud

Kundenerlebbare Funktionen
Tür entriegeln
Tür verriegeln



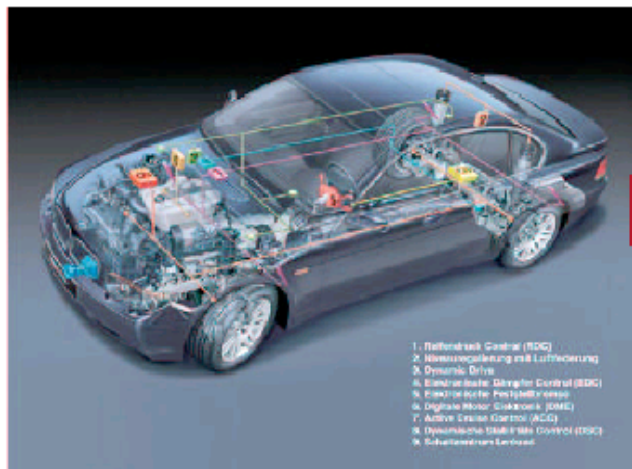
Warum Software beim OEM?



- Ein großer Teil der Funktionen im Fahrzeug hängt direkt oder indirekt mit Software zusammen
- Software bestimmt auch wesentlich das Zusammenspiel zwischen Onboard- und Offboard-Systemen in Produktion und Service
- Die Komplexität der Software ist durch Funktionszuwachs und Integration dramatisch gewachsen
- Die Software wird von einer Vielzahl von Lieferanten erstellt, die Rolle des Systemintegrators hat der OEM
- Software wird zunehmend zu einem der Hauptrisikofaktoren für den Anlauf einer neuen Baureihe
- Hohe wirtschaftliche Risiken durch Rückruf oder Produkthaftungsfälle
- Obwohl die Software im Fahrzeug überwiegend von Lieferanten erstellt wird, muß der OEM die qualitäts- und termingerechte Erstellung und Auslieferung kontrollieren und steuern können.

Der aktuelle Stand

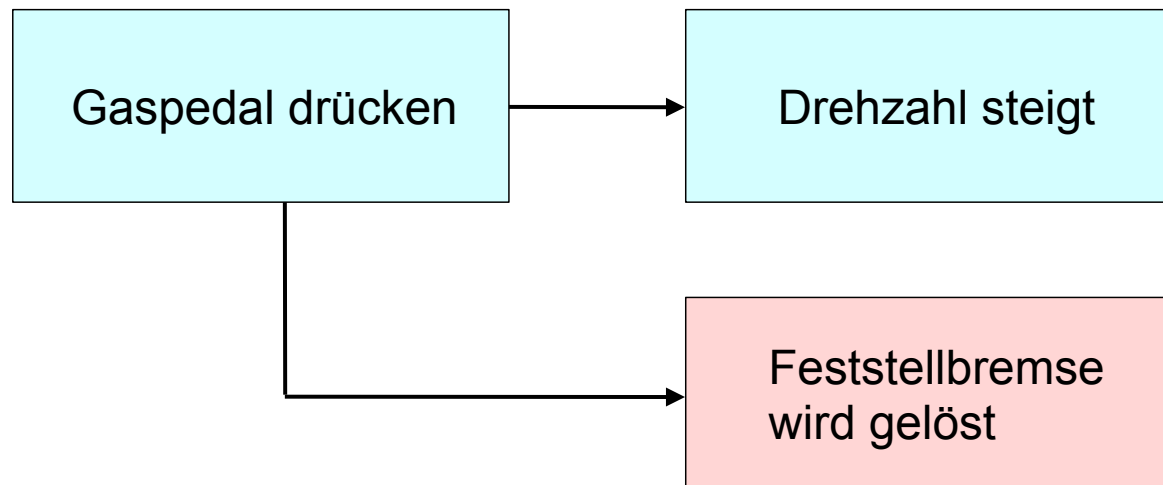
- Bis zu 40% der Herstellkosten eines Fahrzeugs werden heute durch Elektronik/Software bestimmt
- 90% aller Innovation sind getrieben durch Elektronik/Software
- Oberklassefahrzeuge besitzen bis zu 70 ECU's, welche über verschiedene Bussysteme kommunizieren

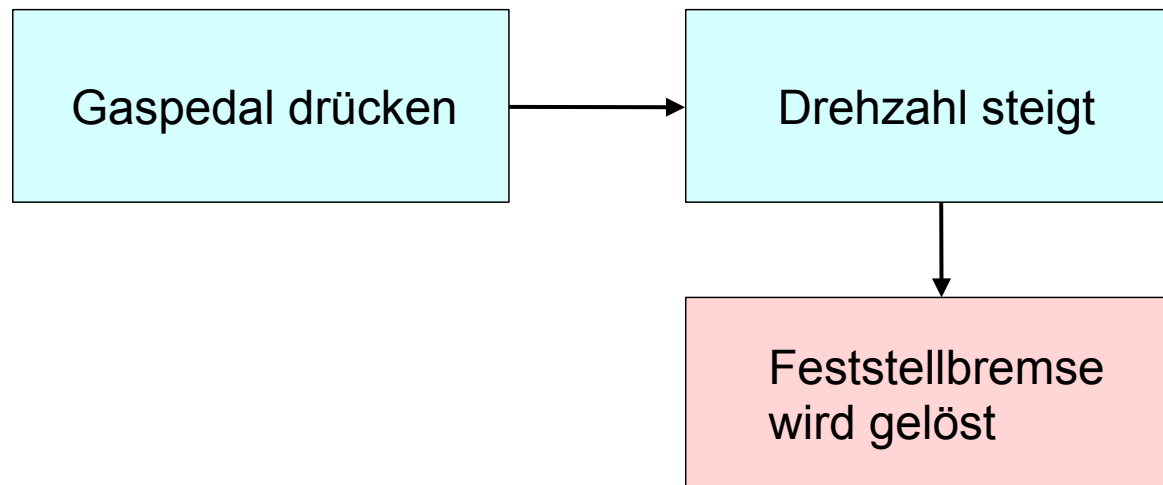


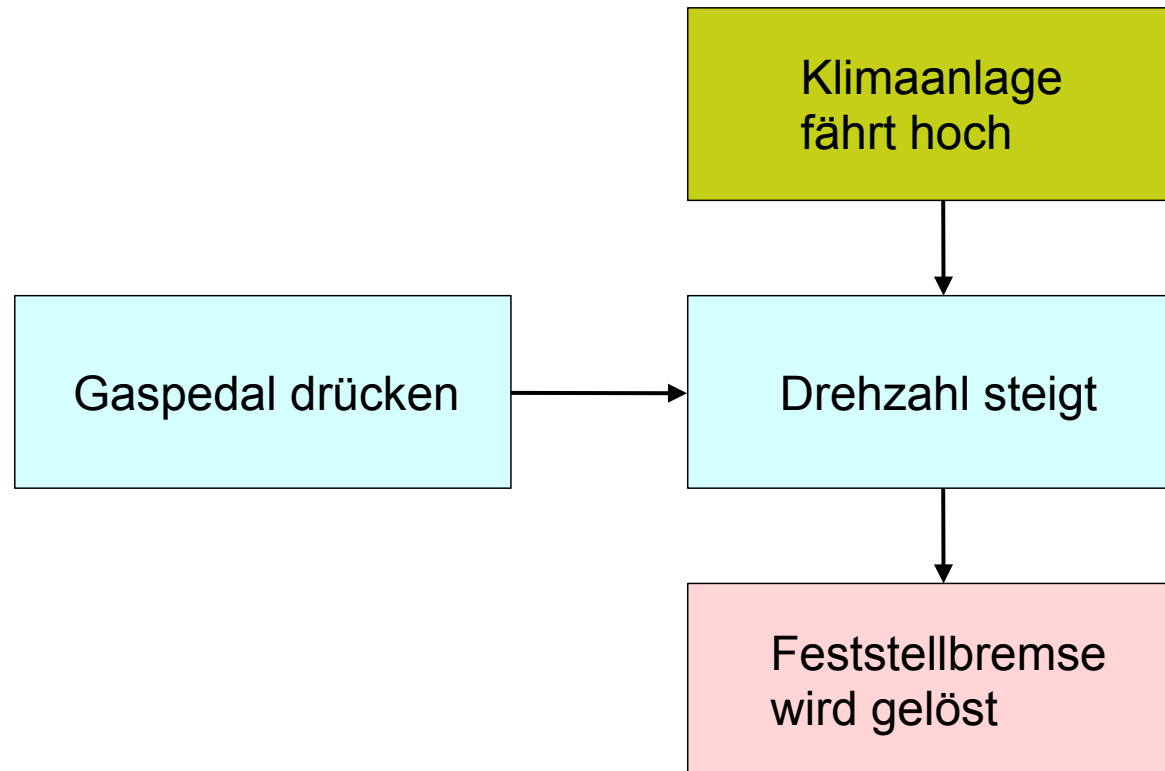
- Steigende Systemkomplexität
- Erhöhte Abhängigkeiten
- Kosten spielen tragende Rolle

- ❑ **Erheblich steigender Softwareanteil erfordert**
 - Strukturierter Software Entwurf
 - UML-Profil für Steuergeräte: Automotive UML !
 - Hardware unabhängigen Funktionsentwurf (seperiere Verhalten von der HW Architektur)
 - Wiederverwendung (Komposition von Teilen)
 - Hardware Plattformen
 - Abbildung der Funktionen auf unterschiedliche HW Plattformen ohne Codeänderung
 - Berechnung und Kommunikation
- ❑ **Mehr Optimierung auf Systemebene (verteilte Steuergeräte)**
- ❑ **Verändert Kooperationsmodell: SW-Module von verschiedenen Zulieferern (Intellectual Property) auf eine oder mehrere HW-Plattformen**
- ❑ **Entwurf für Wiederverwendung und Entwurf mit Wiederverwendung.**

- **sehr gute Kenntnisse in Hardwarearchitekturen (Prozessoren, verteilte Systeme).**
- **exzellente theoretische und praktische Kenntnisse im Entwurf von komplexer Software.**
- **Kenntnisse in der Beschreibung (Modellierung) und in der Analyse komplexer informationstechnischer Systeme.**
- **Verständnis der ingenieurtechnischen Grundlagen der Anwendung.**
- **Grundkenntnisse in Schaltungstechnik und elektrischer Datenerfassung.**



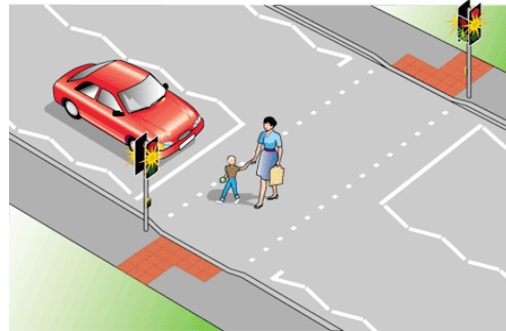




Example

7

Hazard: Unintended vehicle lurch



Safety Goal:

- Cranking the engine by the Micro-Hybrid Stop-Start Feature shall not contribute to vehicle movement (transfer torque to wheels) in other than vehicle pull-away maneuvers.



Functional and Technical Safety Concept

SW Safety Requirement:

- If the starter command is CRANK and the gear state is not NEUTRAL then the starter command shall be reset



Research & Advanced Engineering

Thomas Rambow [19 Nov 2009]

Ladestrategien

- Mindestladung der Batterie erhalten
- Ab definierter Motordrehzahl laden
- Nur Bremsenergie laden
- Konstanter Ladestrom
- ...

Zuschaltung Elektromotor

- Bis Richtgeschwindigkeit
- Ortsbezogen
- Ladungsbezogen
- Booster (siehe SPIEGEL 13.02.2010)
- ...

Kein Rosenmontagsscherz



if <condition>

then

....

goto L2;

....

L1:

....

else

....

L2:

....

goto L1;

....

end if;

while <condition>

do

...

goto L;

-- irgendwo ausserhalb der

-- Schleife

...

end while;

1. Motivation und Überblick

1. Bedeutung der Automobilindustrie für die deutsche Volkswirtschaft

2. Motivation Automotive Software Engineering

3. GI-Fachgruppe Automotive Software Engineering

4. Software Entwicklungsprozess

5. Standards zur Softwareentwicklung

6. Software Architekturen

7. AUTOSAR

8. Lessons Learned

Ziele der Fachgruppe (1)



- <http://www.gi-ev.de/fachbereiche/softwaretechnik/ase/>

- **Motivation**

- Das Thema „Automotive Software Engineering“ ist insbesondere in Deutschland und Europa von ständig wachsender Bedeutung, die Automobilindustrie ist bereits heute ein entscheidender Innovationstreiber im Bereich der eingebetteten Systeme.
- Langfristig sind intensive Anstrengungen erforderlich, um den deutschen Wettbewerbsvorsprung bei software-basierten Innovationen im Automobil zu halten.
- Die Automobilindustrie wird deswegen in weiter zunehmendem Maße ein wichtiger Anwender der Informatik und ein wichtiger Arbeitgeber für Informatiker sein.
- Das Gebiet wirft spezifische interessante Forschungsfragen auf.
- Die stärkere Berücksichtigung des Gebiets in der Lehre ist zu erreichen.

■ Vorgeschichte

- Workshops zum Thema „Automotive Software Engineering“ auf der ICSE und der GI-Jahrestagung (jeweils seit 2003) ermöglichten einen intensiven Austausch zwischen der Automobilindustrie und der wissenschaftlichen Community.
- Insbesondere die Darstellung und Diskussion des State-of-the-Art im Software-Engineering in der Automobilindustrie lieferte wertvolle Anforderungen an die zukünftigen Handlungsbedarfe und Forschungsschwerpunkte auf diesem Gebiet, und aktuelle Forschungsergebnisse konnten einer breiten automobilen Öffentlichkeit vorgestellt werden.
- Die Fachgruppe „Automotive Software Engineering (ASE)“ wurde auf Antrag von Dr. Klaus Grimm im Februar 2005 vom Präsidium Gesellschaft für Informatik eingerichtet. Die FG ist dem Fachbereich Softwaretechnik zugeordnet. Sie hatte am 21. September 2005 im Rahmen der GI-Jahrestagung in Bonn ihre Gründungsversammlung.
- Ziel der Fachgruppe „Automotive Software Engineering“ im Fachbereich Softwaretechnik der Gesellschaft für Informatik (GI) ist der intensive fachliche Austausch zwischen der Automobilindustrie und der wissenschaftlichen Community.

- <http://www1.gi-ev.de/fachbereiche/softwaretechnik/ase/>
- Die Fachgruppe veranstaltet regelmässig den Workshop „Automotive Software Engineering“ im Rahmen der Jahrestagungen der Gesellschaft für Informatik.
 - 30. September 2010 in Leipzig
Organisation: Prof. Dr. Jörn Schneider, FH Trier
 - 28. September 2009 in Lübeck
Organisation: Dr. Christian Allmann, Audi Electronics Venture GmbH
 - 11. September 2008 in München
Organisation: Dr. Reinhard Stolle, BMW Car IT
 - 27. September 2007 in Bremen
Organisation: Dr. Michael Reinfrank, Siemens VDO
 - 5. Oktober 2006 in Dresden
Organisation: Dr. Michael Daginnus, Volkswagen AG
 - 21. September 2005 in Bonn
Organisation: Dr. Thomas Kropf, Robert Bosch GmbH
 - 23. September 2004 in Ulm
Organisation: Dr. Bernhard Hohlfeld, DaimlerChrysler AG
 - 30. September 2003 in Frankfurt
Organisation: Dr. Alexandre Saad, BMW Group

1. Motivation und Überblick

1. Bedeutung der Automobilindustrie für die deutsche Volkswirtschaft

2. Motivation Automotive Software Engineering

3. GI-Fachgruppe Automotive Software Engineering

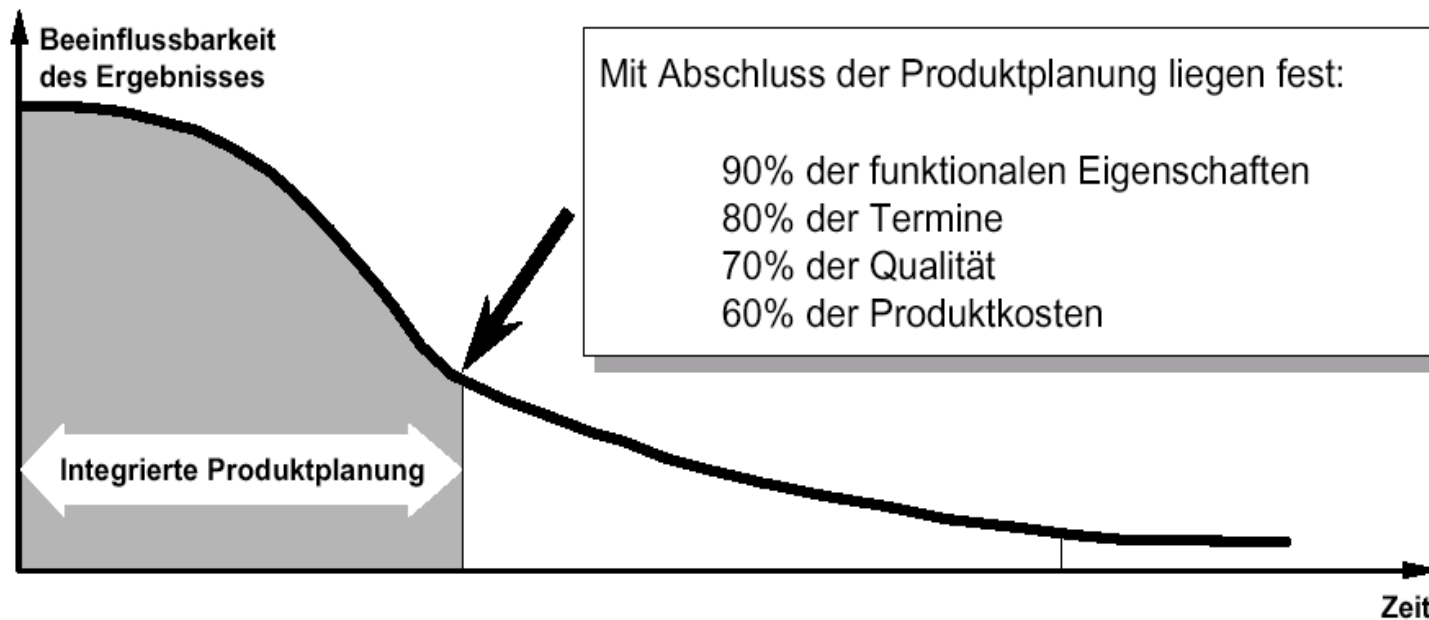
4. Software Entwicklungsprozess

5. Standards zur Softwareentwicklung

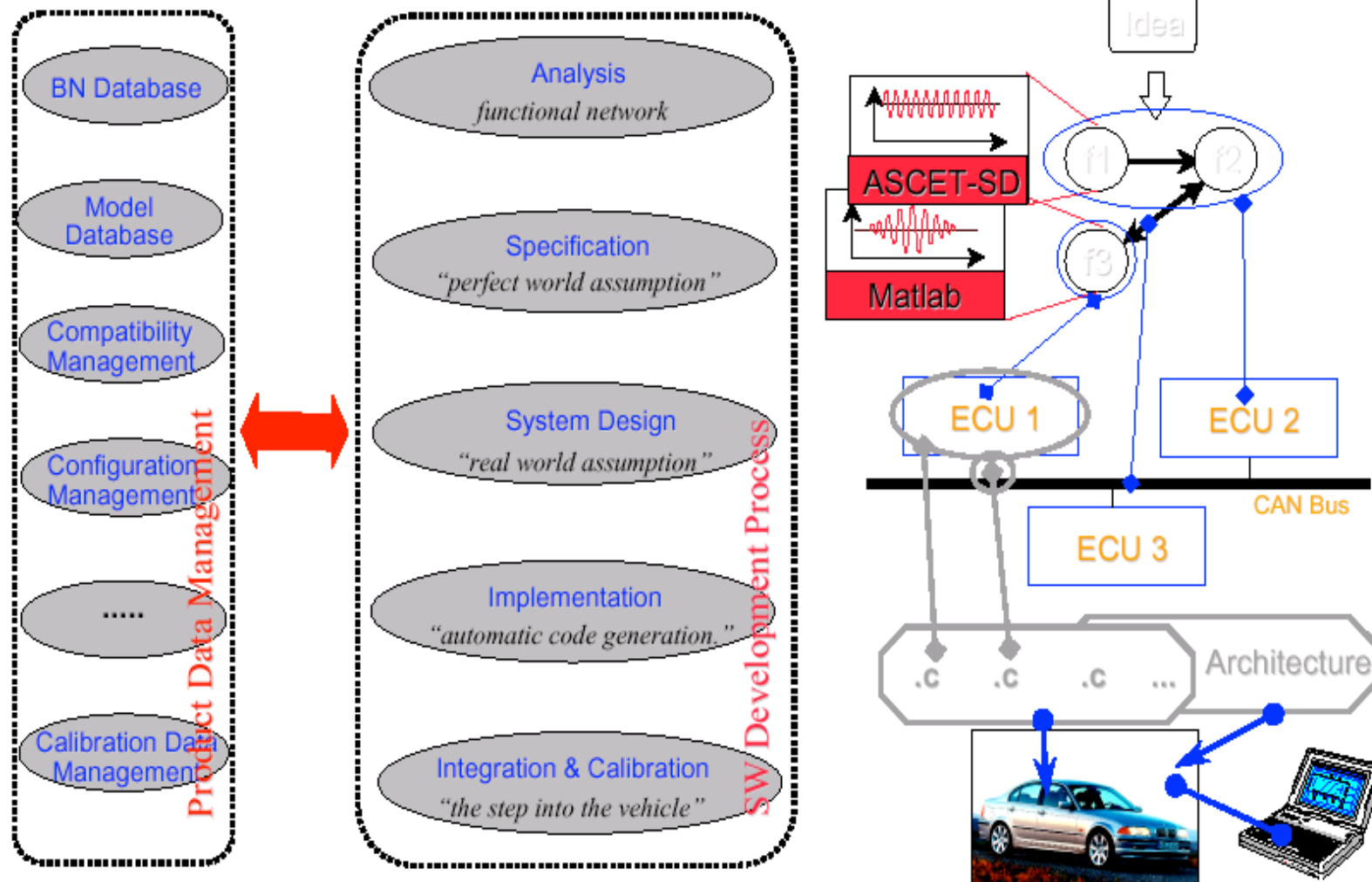
6. Software Architekturen

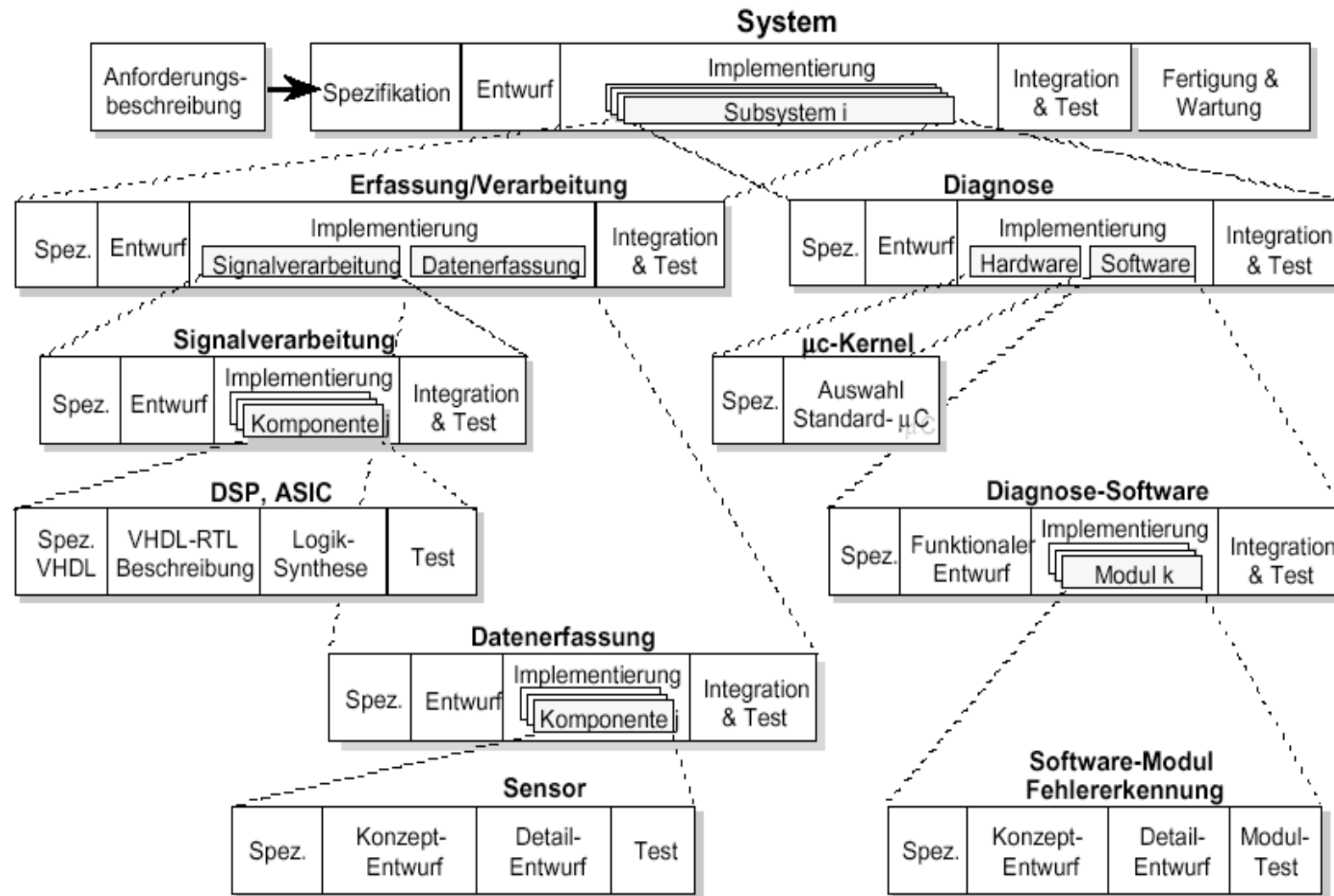
7. AUTOSAR

8. Lessons Learned

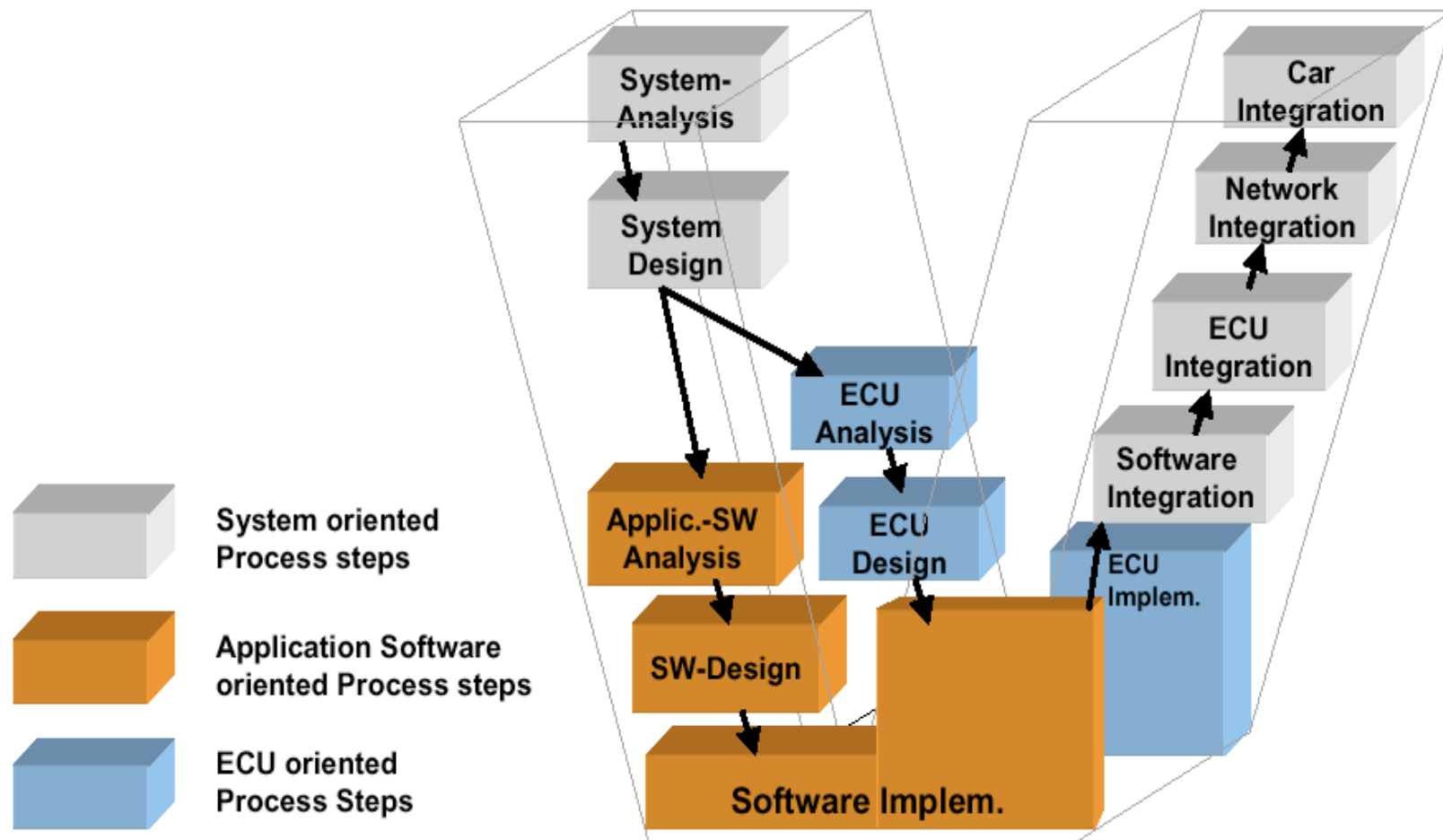


System Planung			System Realisierung				System Einsatz	
System Strategie Planung	System Profil Planung	System Konzept Planung	System Entwurf	Komponenten Entwicklung und Test	System Integration und Test	Fertigungs-Einführung	Fertigungs-Durchführung	System-Betreuung

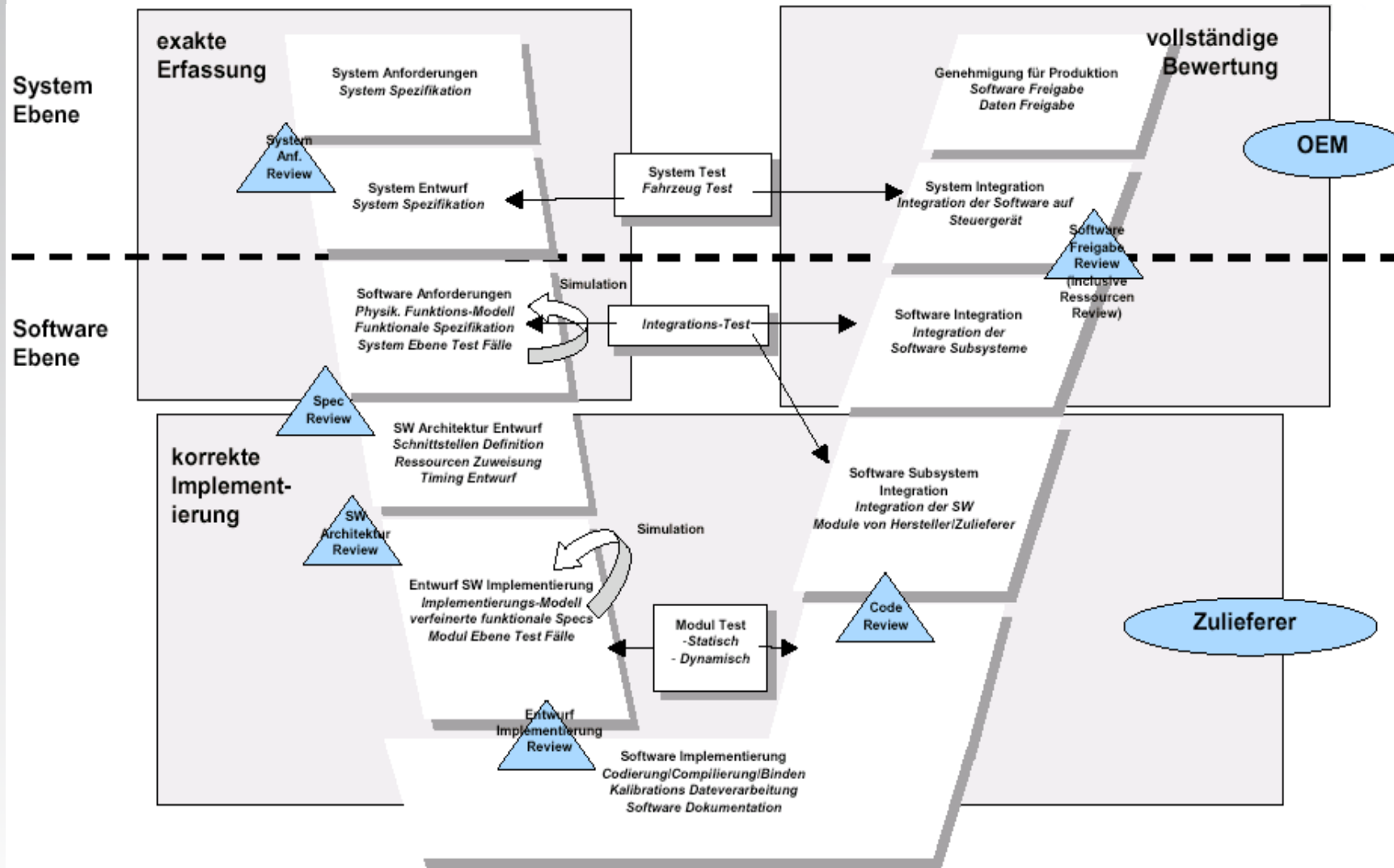




Software Entwicklungsprozess Abbildung auf V-Modell

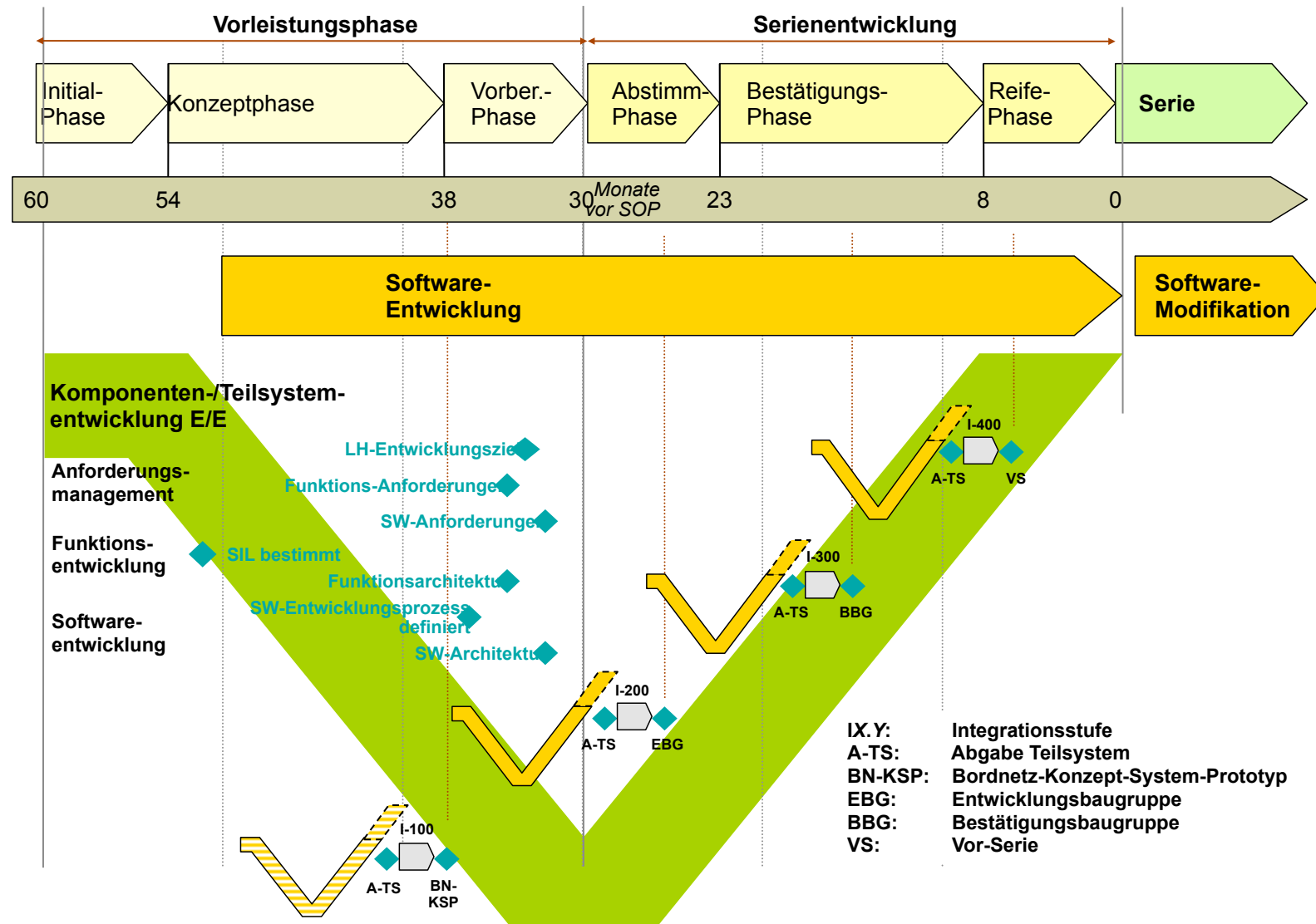


Detaillierung V-Modell



Software Entwicklungsprozess

Einordnung in Produktentstehungsprozess



Software Entwicklungsprozess

Einordnung in Produktentstehungsprozess



■ Komponenten-/Teilsystementwicklung E/E

- LH-Entwicklungsziel
- Funktions-Anforderungen
- SW-Anforderungen

■ Funktionsentwicklung

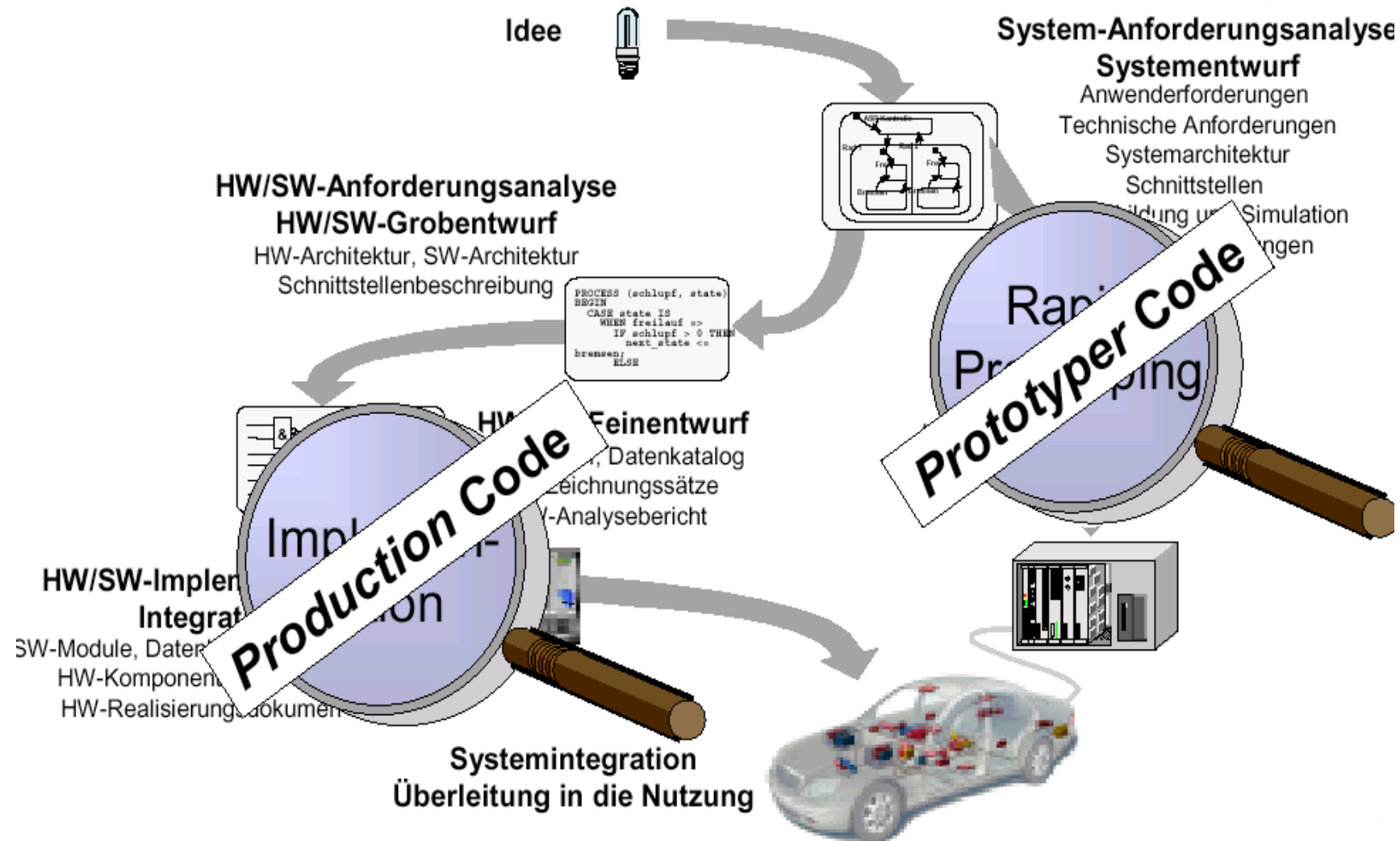
- SIL bestimmt
- Funktionsarchitektur

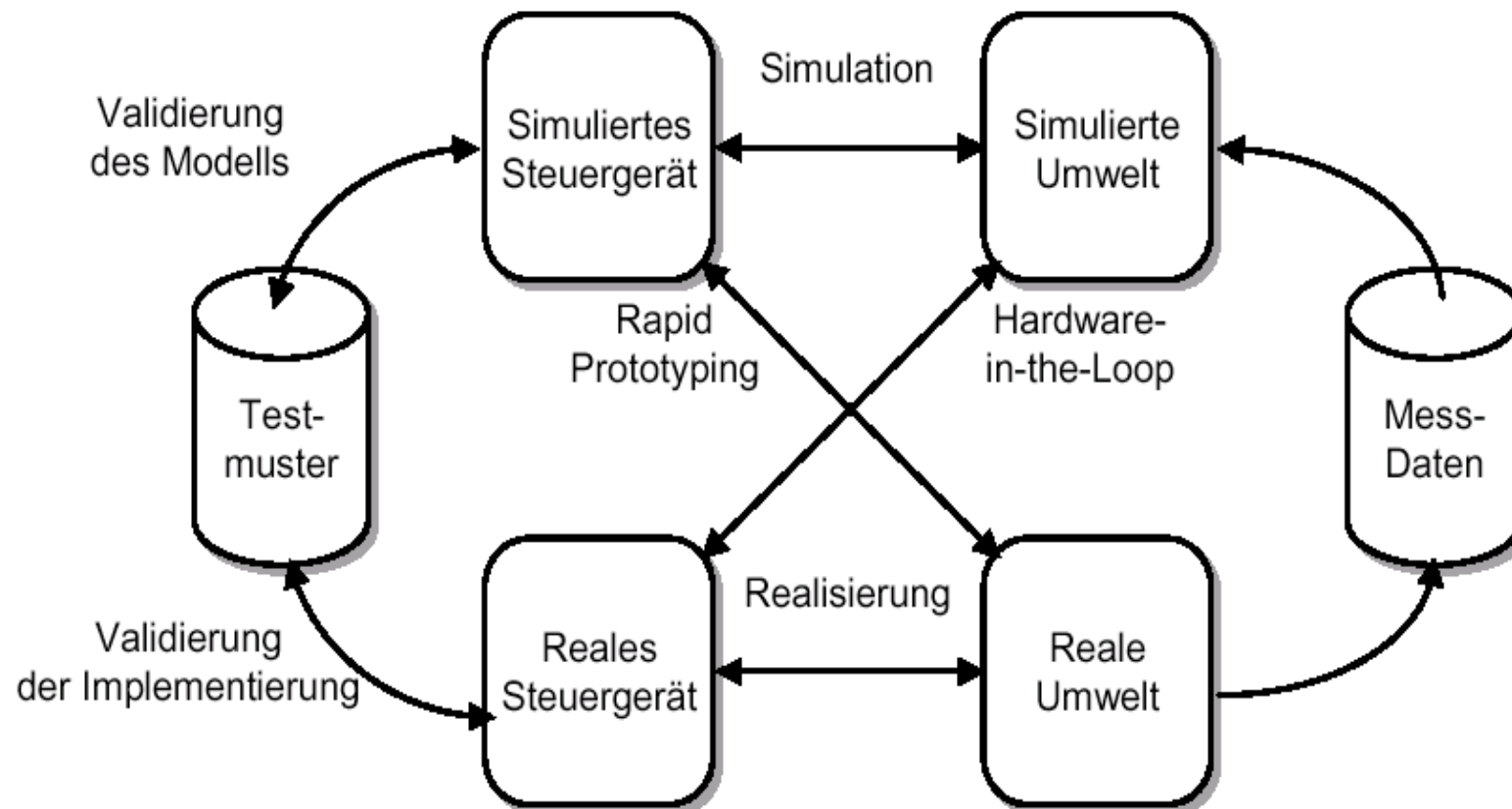
■ Softwareentwicklung

- SW-Entwicklungsprozess definiert
- SW-Architektur

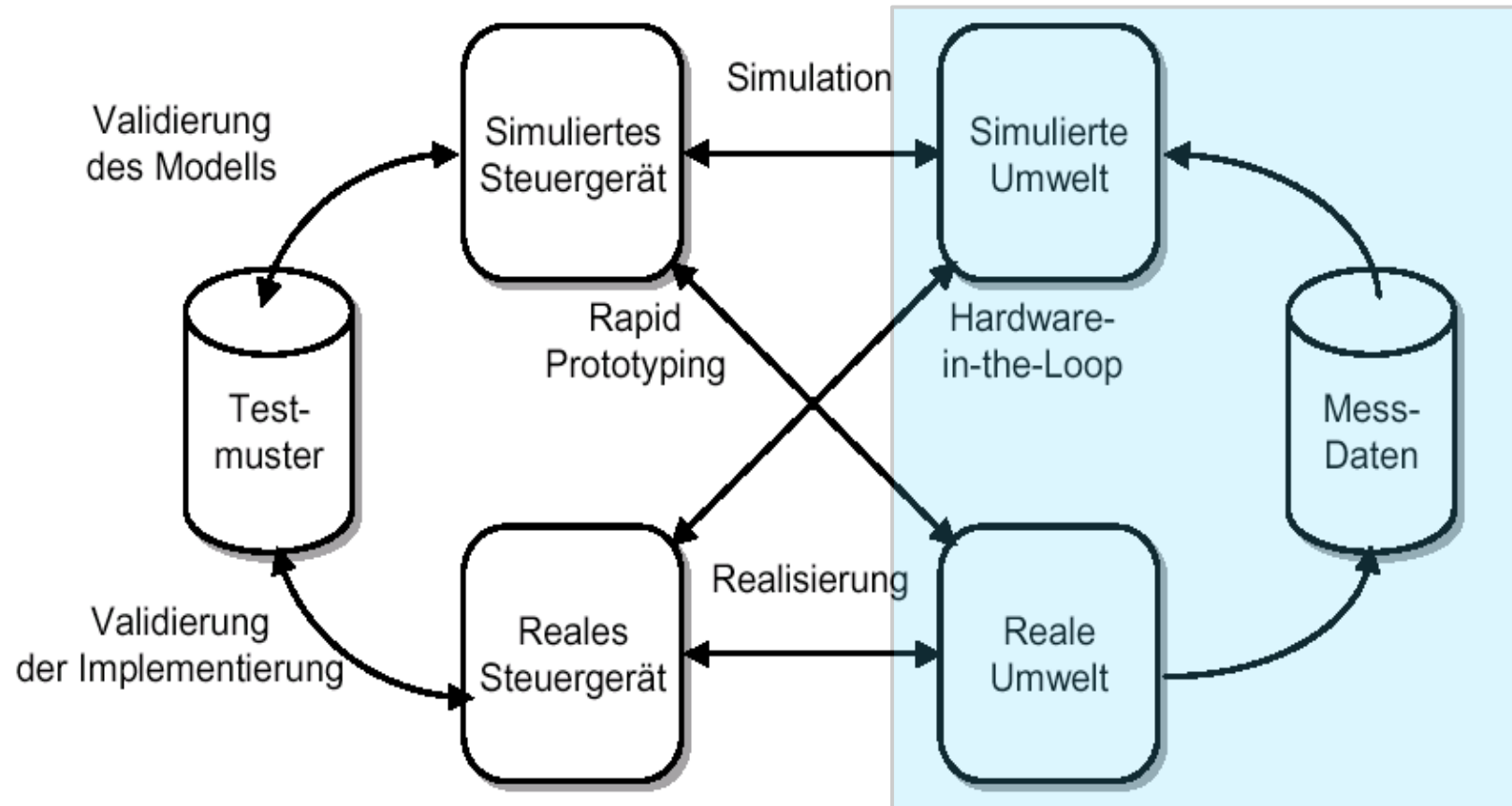
■ Integrationsstufen

- I-100 BN-KSP: Bordnetz-Konzept-System-Prototyp
- I-200 EBG: Entwicklungsbaugruppe
- I-300 BBG: Bestätigungsbaugruppe
- I-400 VS: Vor-Serie

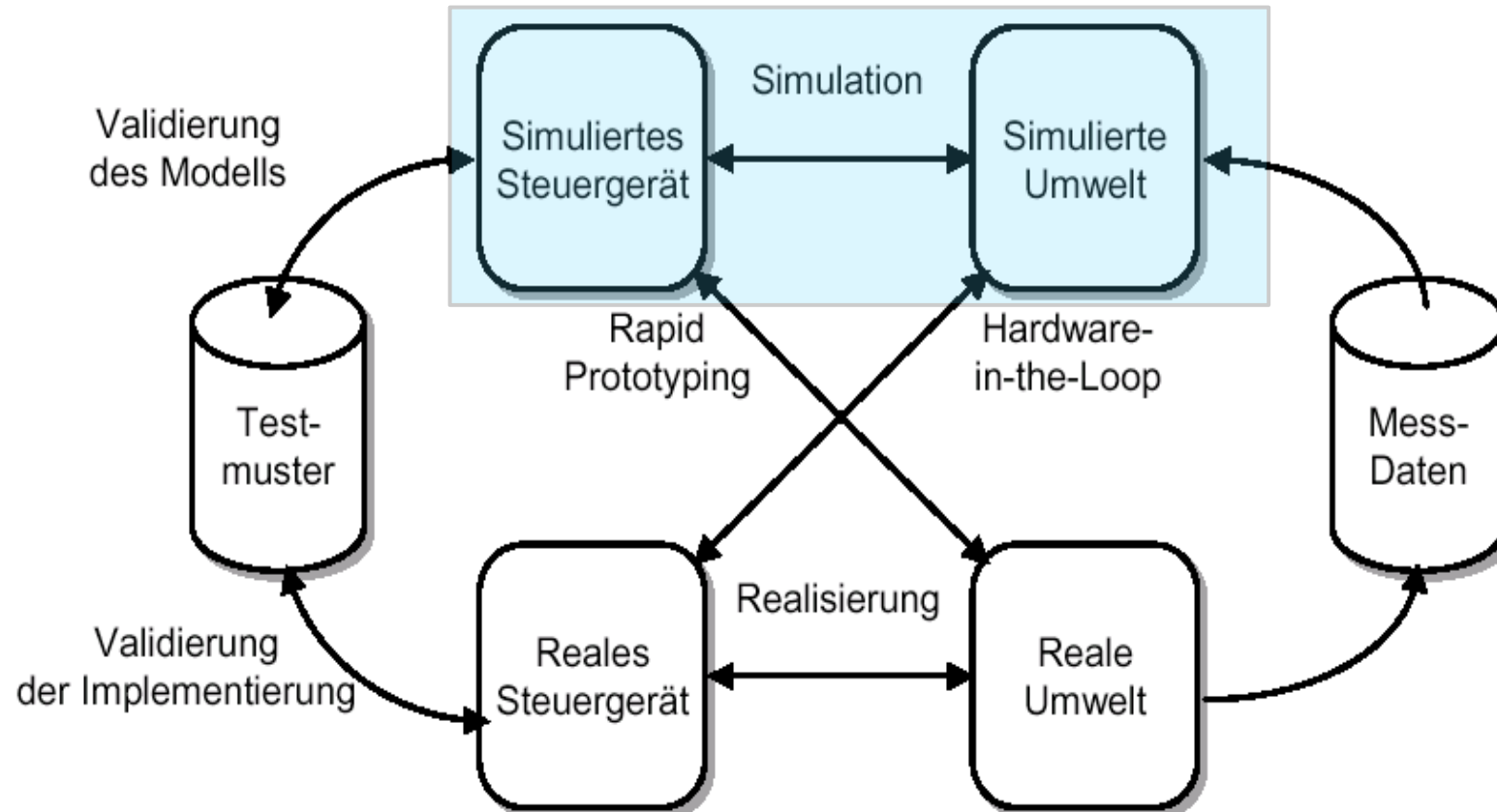




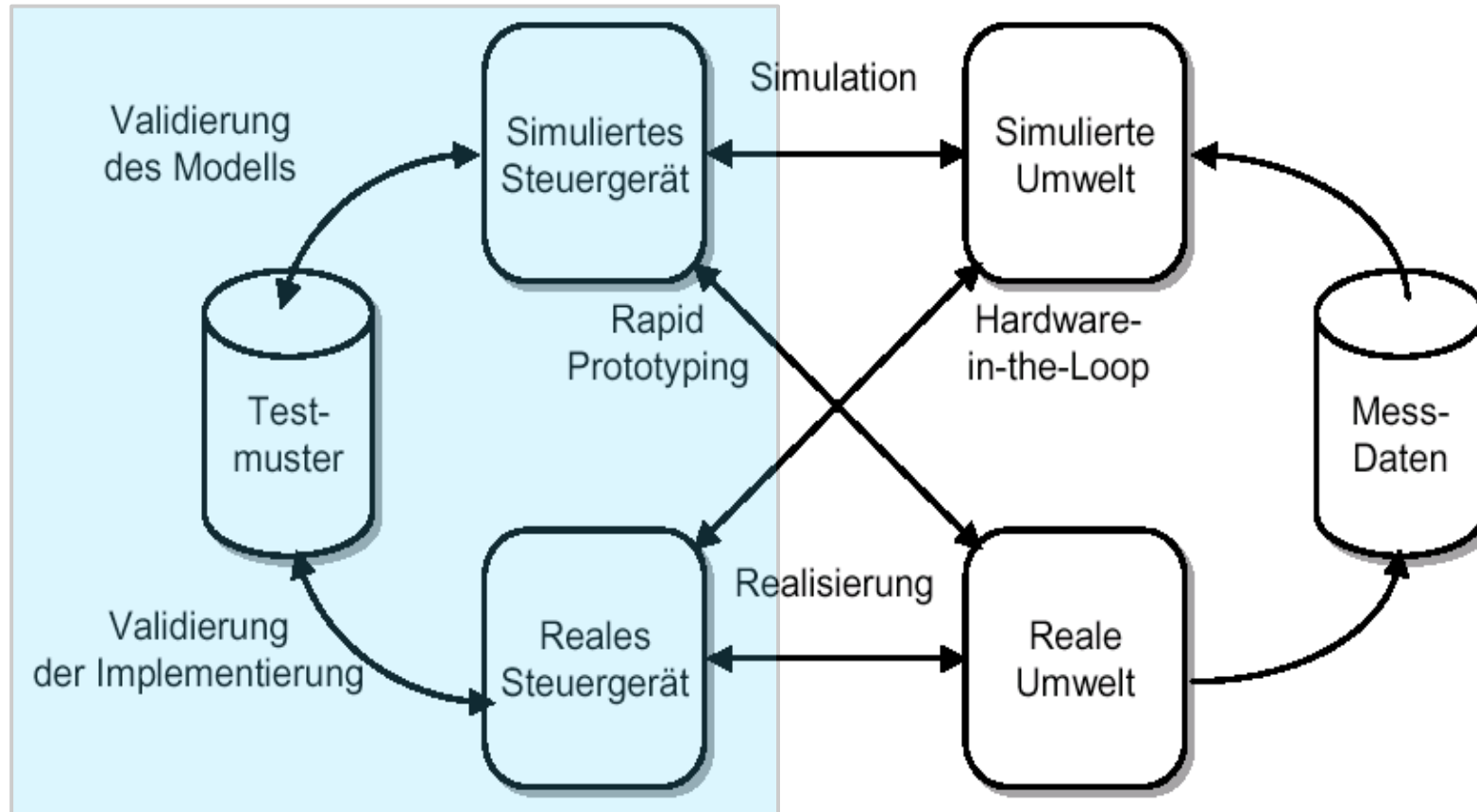
- ☐ **Rapid Prototyping für frühe Tests in realer Umgebung**
- ☐ **Hardware-in-the-Loop für umfangreiche Tests des Steuergerätes**



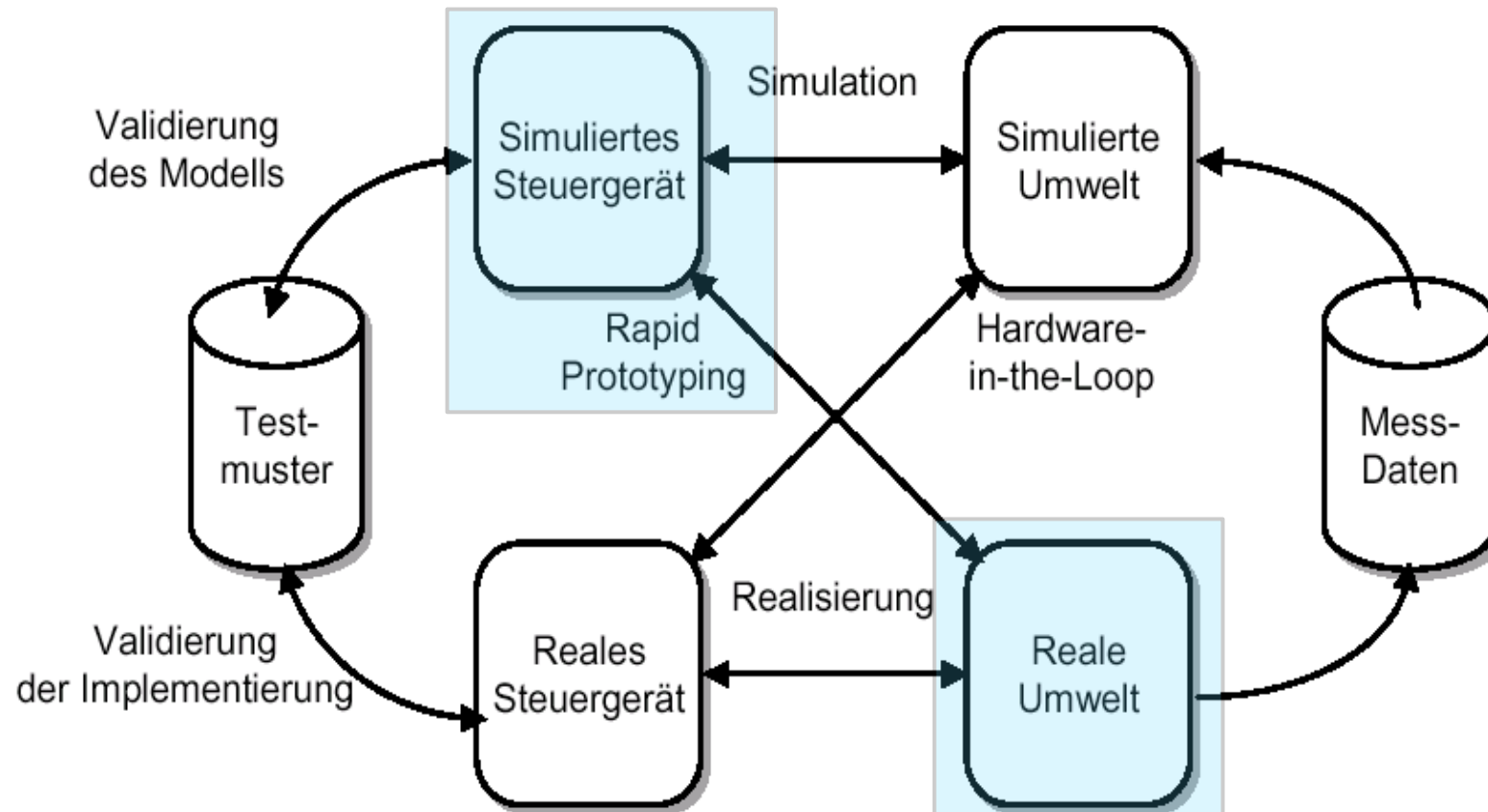
- ❑ **Rapid Prototyping für frühe Tests in realer Umgebung**
- ❑ **Hardware-in-the-Loop für umfangreiche Tests des Steuergerätes**



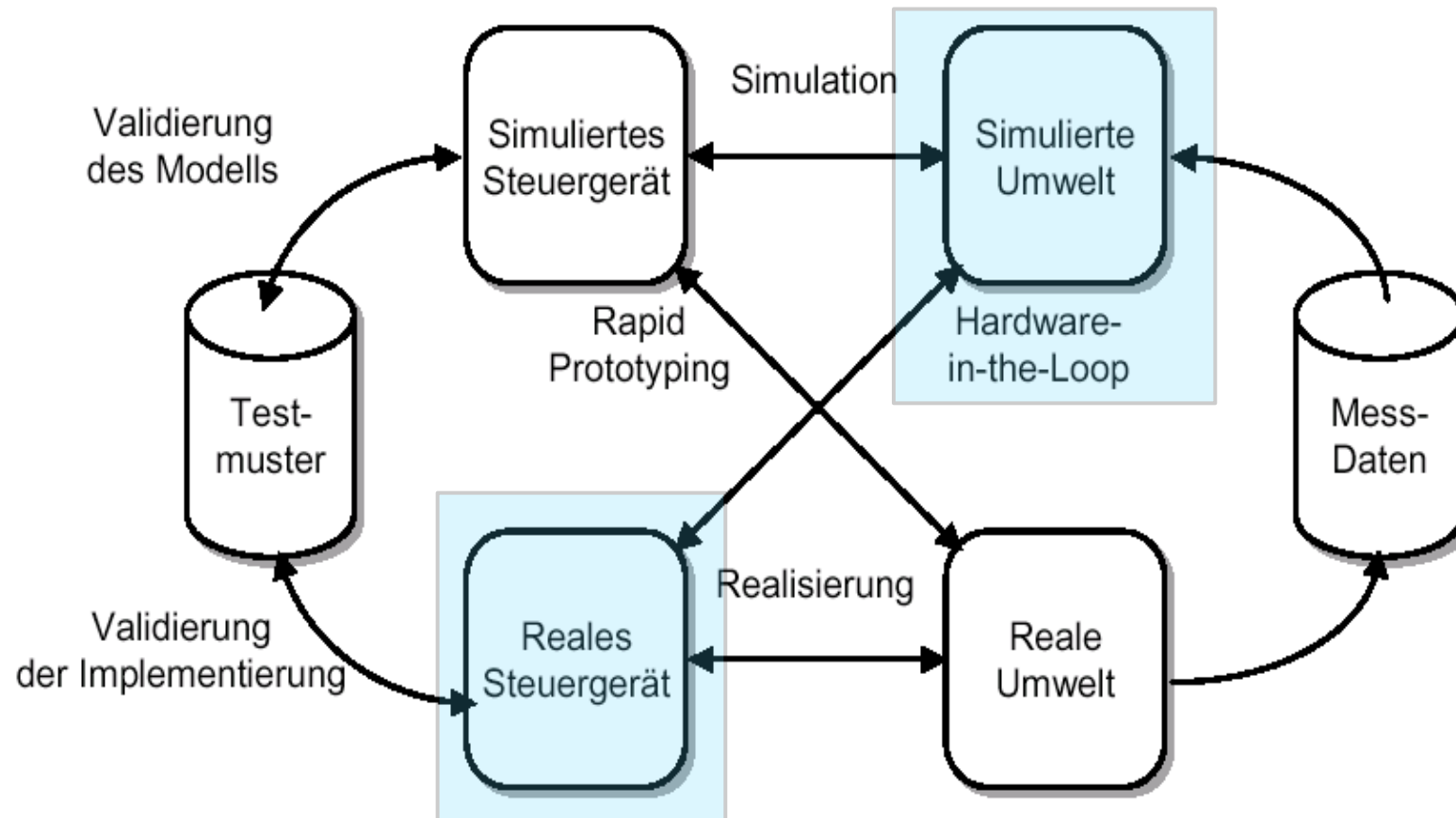
- ☐ Rapid Prototyping für frühe Tests in realer Umgebung
- ☐ Hardware-in-the-Loop für umfangreiche Tests des Steuergerätes



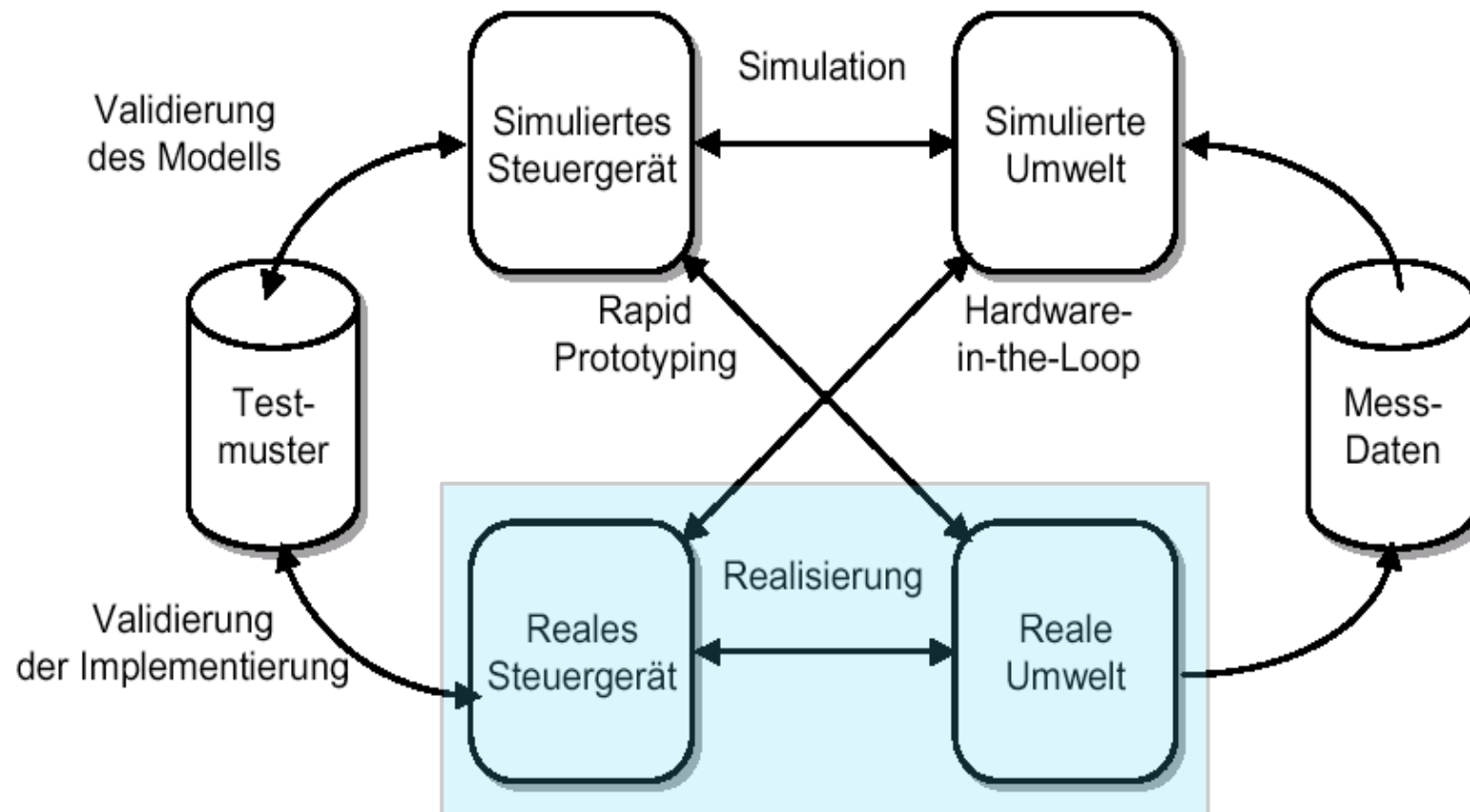
- ☐ Rapid Prototyping für frühe Tests in realer Umgebung
- ☐ Hardware-in-the-Loop für umfangreiche Tests des Steuergerätes



- ☐ **Rapid Prototyping für frühe Tests in realer Umgebung**
- ☐ **Hardware-in-the-Loop für umfangreiche Tests des Steuergerätes**



- ☐ **Rapid Prototyping für frühe Tests in realer Umgebung**
- ☐ **Hardware-in-the-Loop für umfangreiche Tests des Steuergerätes**



- ☐ Rapid Prototyping für frühe Tests in realer Umgebung
- ☐ Hardware-in-the-Loop für umfangreiche Tests des Steuergerätes

Testvarianten im Integrationsprozess



- MIL Model in the Loop
- SIL Software in the Loop
- PIL Prozessor in the Loop
- HIL Hardware in the Loop
 - K-HIL Komponente
 - S-HIL System
- VIL Vehicle in the Loop
- EFP Erweiterte Funktionsprüfung

- ❑ **Erheblich steigender Softwareanteil erfordert**
 - Strukturierter Software Entwurf
 - UML-Profil für Steuergeräte: Automotive UML !
 - Hardware unabhängigen Funktionsentwurf (separiere Verhalten von der HW Architektur)
 - Wiederverwendung (Komposition von Teilen)
 - Hardware Plattformen
 - Abbildung der Funktionen auf unterschiedliche HW Plattformen ohne Codeänderung
 - Berechnung und Kommunikation
- ❑ **Mehr Optimierung auf Systemebene (verteilte Steuergeräte)**
- ❑ **Verändert Kooperationsmodell: SW-Module von verschiedenen Zulieferern (Intellectual Property) auf eine oder mehrere HW-Plattformen**
- ❑ **Entwurf für Wiederverwendung und Entwurf mit Wiederverwendung.**

- **sehr gute Kenntnisse in Hardwarearchitekturen (Prozessoren, verteilte Systeme).**
- **exzellente theoretische und praktische Kenntnisse im Entwurf von komplexer Software.**
- **Kenntnisse in der Beschreibung (Modellierung) und in der Analyse komplexer informationstechnischer Systeme.**
- **Verständnis der ingenieurtechnischen Grundlagen der Anwendung.**
- **Grundkenntnisse in Schaltungstechnik und elektrischer Datenerfassung.**

1. Motivation und Überblick

1. Bedeutung der Automobilindustrie für die deutsche Volkswirtschaft

2. Motivation Automotive Software Engineering

3. GI-Fachgruppe Automotive Software Engineering

4. Software Entwicklungsprozess

5. Standards zur Softwareentwicklung

6. Software Architekturen

7. AUTOSAR

8. Lessons Learned

■ CMMI

- Bewertung durch Level, sehr generisch
- Beschreibung was zu geschehen hat, aber nicht wie es zu geschehen hat

■ ISO 15504 SPICE / Automotive SPICE

- Level direkt vergleichbar mit CMMI

■ ISO 12207 AMD 1

- international
- sehr generisch
- Detaillierung erforderlich

■ V-Modell 97, V-Modell XT

- national
- konkret
- anleitend
- beschreibt Dokumente

■ VDA AK 13

- gute Definitionen
- informativ
- unverbindlich

■ IEC 61508 /

- weite Verbreitung
- Generisch, branchenspezifische Detaillierung erforderlich

■ EN 50128 (Eisenbahn)

- pragmatisch
- benutzerfreundlich

■ ISO CD 26262 (Automotive)

- In Einführung

■ RTCA DO-178B (Luftfahrt)

- konkret
- präziser als IEC 61508

■ Def Stan 00-55

- klares Rollenkonzept

■ MISRA Guidelines

- Spezifika des Automobilsektors

ISO WD 26262 (Automotive E/E Sicherheitsengineering)



■ Ziele

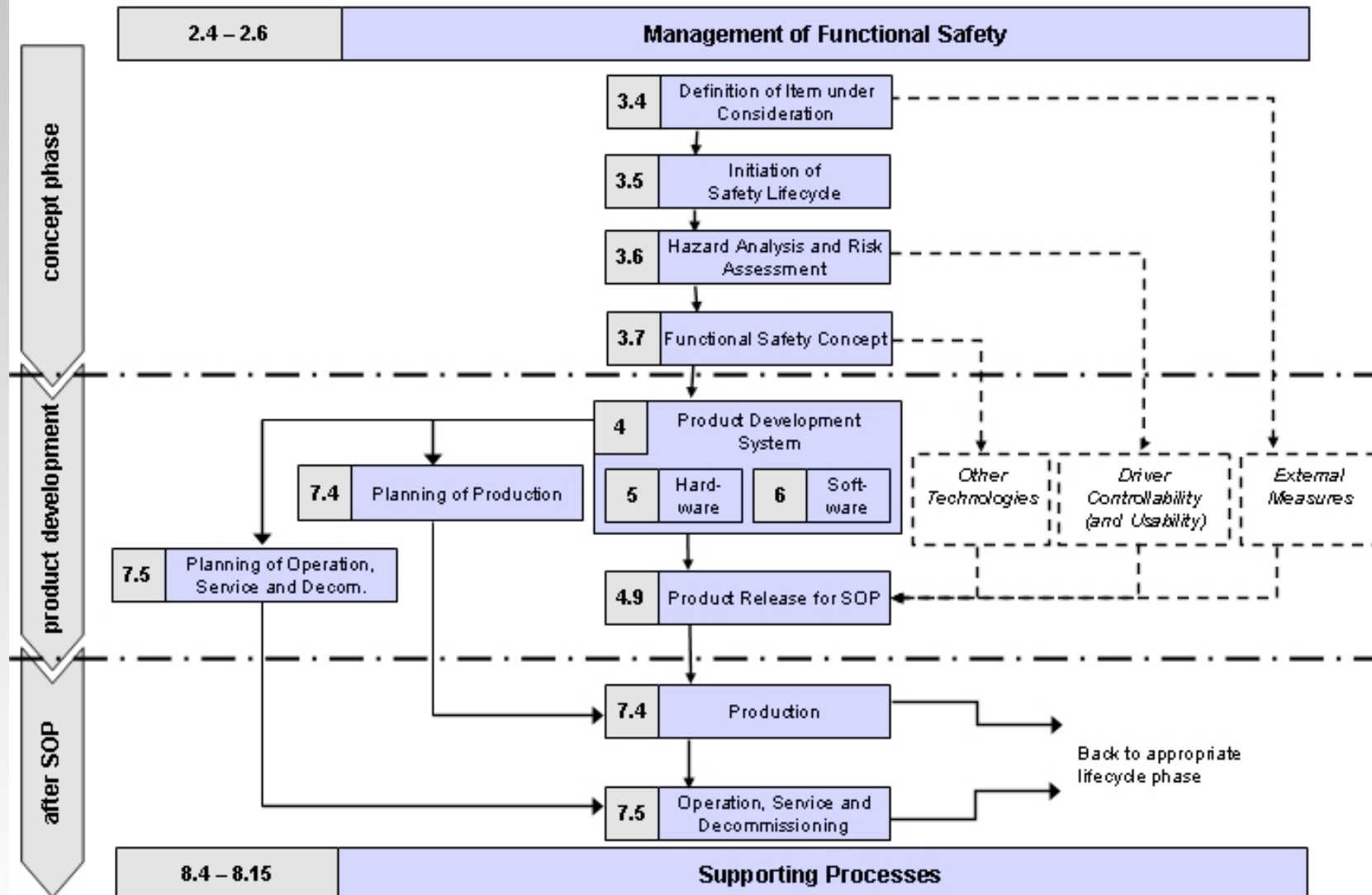
- Die ISO WD 26262 ist ein Normentwurf von Vertretern der Automobilindustrie für Sicherheit elektronischer Fahrzeugsysteme.

■ Nächste Folie:

- Gesamter Lebenszyklus nach ISO WD 26262

■ Quelle

- [http://keng.ke.ohost.de/ISO WD26262\(AutomotiveE.ESicherheitsengineering\).html](http://keng.ke.ohost.de/ISO WD26262(AutomotiveE.ESicherheitsengineering).html)



■ Ziele

- verbindlicher Standard für die Entwicklung von Steuergeräte-Software beim OEM wie bei den Lieferanten;
- Berücksichtigung von Sicherheitsanforderungen;
- Formulierungen von Prozessen, Aufgaben und Produkten;
- Gewährleistung des internationalen Stands der Technik.

■ Vorgehen

- Gewährleistung des internationalen Stands der Technik durch Abstützung auf internationale Normen, wie z. B. CMMI, ISO 12207 und IEC 61508;
- Anpassung und ggf. Ergänzung an die OEM-Belange.

■ Ergebnis

- Wichtige Basis für das SW-Qualitätsmanagement
- Planungsgrundlage für SW-Entwicklungsprojekte
- Verbesserung der SW-Qualität

■ Beispiel

- BMW Group Standard embedded Software (GSeSW)

■ Quelle

- http://www.dspace.de/ww/de/gmb/home/company/events/4ankon/bmw_riedesser_rissling.cfm





- ist eine Anpassung der internationalen Normen an die Rahmenbedingungen beim OEM und den Lieferanten;
- muss für die Anwendung in Projekten beim OEM und den Lieferanten weiter konkretisiert werden;
- fasst Anforderungen an die Software-Entwicklung aus unterschiedlichen Quellen zusammen;
- stellt Anforderungen an die eigentliche Software-Entwicklung und Anforderungen an begleitende Prozesse;
- gewährleistet den internationalen Stand der Technik.

Der GSeSW

- beschreibt keine Systementwicklung;
- gibt keine technischen Lösungen vor;
- ist keine vollständige Sicherheitsnorm, sondern deckt nur die Anforderungen an die Software ab.

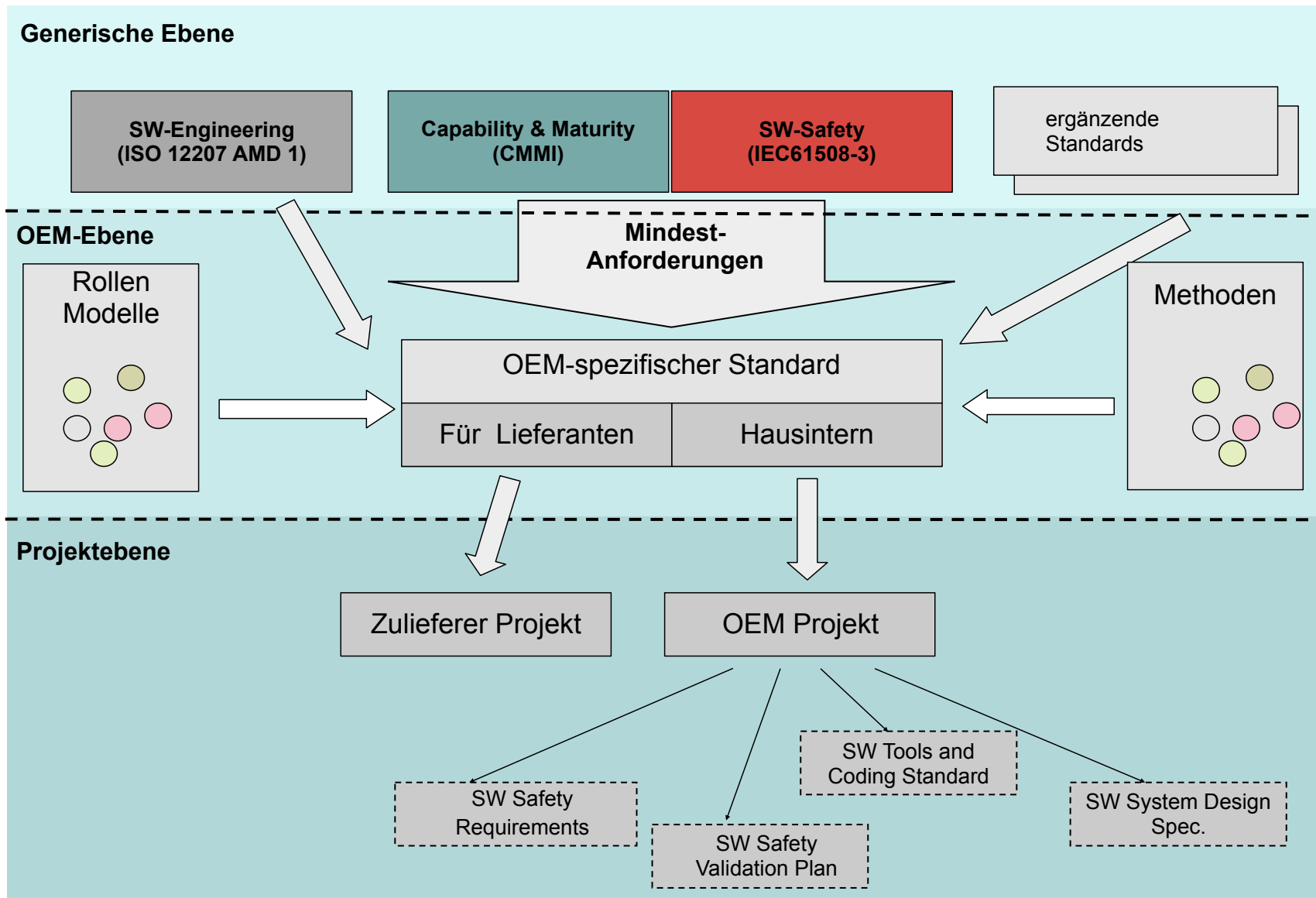
1. Application area 2. Normative links 3. Definitions/abbreviations 4. Structure of this standard		Inform- ational
5. Key processes		
5.1 Project preparations		
5.2 Software development		
5.3 Software modification		
6. Supporting processes		
6.1 Project management		
6.2 Filling of roles		
6.3 Documentation		
6.4 Configuration management		
6.5 Quality assurance		
6.6 Verification process		
6.7 Test process		
6.8 Carrying out reviews		
6.9 Problem resolution process		
7. Organizational processes		
7.1 Selection and qualification of SW tools		7.3 Continual improvement
7.2 Selection and qualification of SW libraries		7.4 Training and qualification
Appendix		
A. Guideline for the selection of methods and measures		
B. Assigning the SIL to the software		C. Literature

Each chapter of the “key processes” section relates to chapters of ISO 12207 and IEC 61508.

- **ist eine Anpassung der internationalen Normen an die Rahmenbedingungen beim OEM und den Lieferanten;**
- muss für die Anwendung in Projekten beim OEM und den Lieferanten weiter konkretisiert werden;
- **fasst Anforderungen an die Software-Entwicklung aus unterschiedlichen Quellen zusammen;**
- stellt Anforderungen an die eigentliche Software-Entwicklung und Anforderungen an begleitende Prozesse;
- gewährleistet den internationalen Stand der Technik.

Der GSeSW

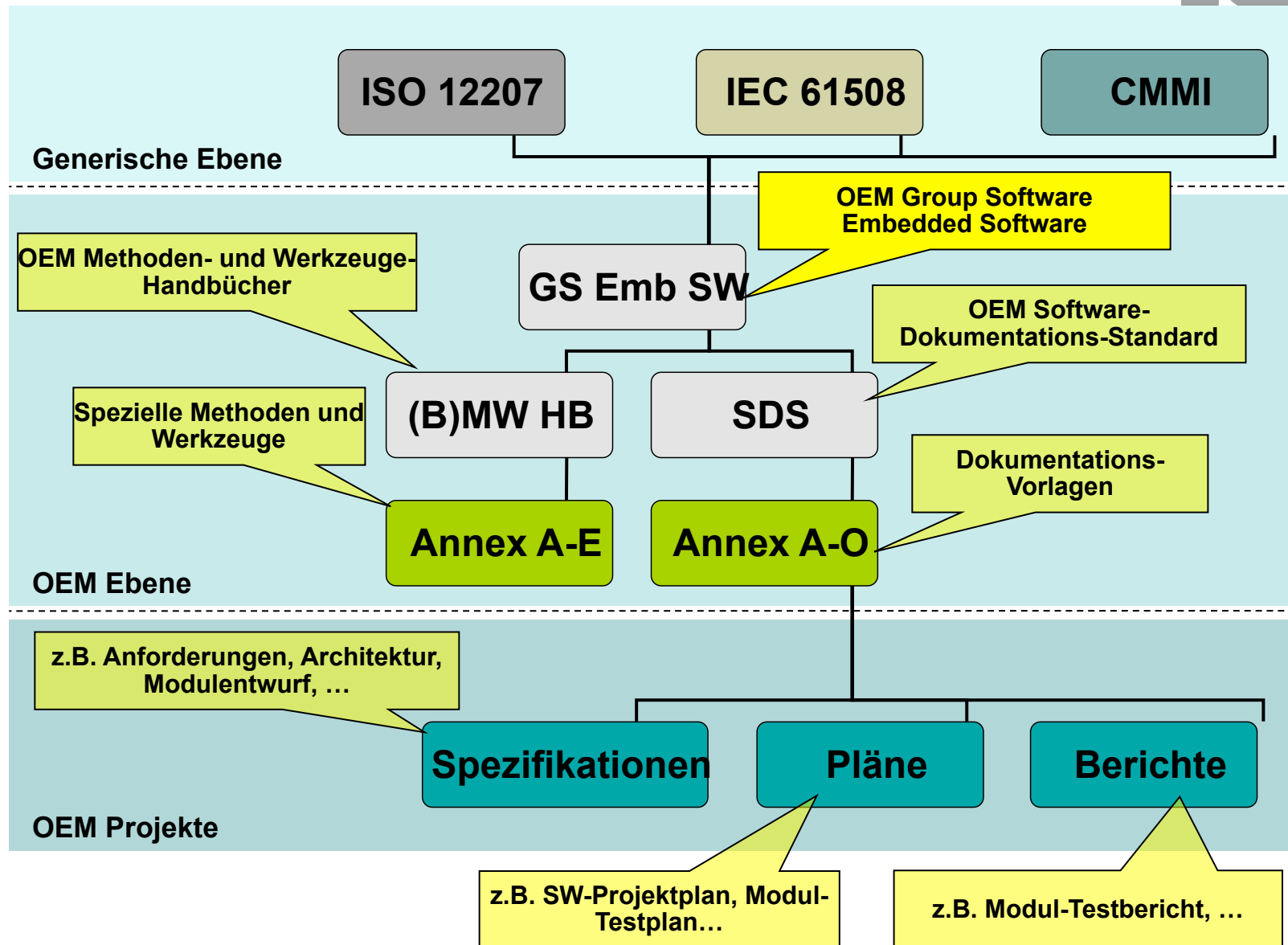
- beschreibt keine Systementwicklung;
- gibt keine technischen Lösungen vor;
- ist keine vollständige Sicherheitsnorm, sondern deckt nur die Anforderungen an die Software ab.



- ist eine Anpassung der internationalen Normen an die Rahmenbedingungen beim OEM und den Lieferanten;
- **muss für die Anwendung in Projekten beim OEM und den Lieferanten weiter konkretisiert werden;**
- fasst Anforderungen an die Software-Entwicklung aus unterschiedlichen Quellen zusammen;
- **stellt Anforderungen an die eigentliche Software-Entwicklung und Anforderungen an begleitende Prozesse;**
- gewährleistet den internationalen Stand der Technik.

Der GSeSW

- beschreibt keine Systementwicklung;
- gibt keine technischen Lösungen vor;
- ist keine vollständige Sicherheitsnorm, sondern deckt nur die Anforderungen an die Software ab.



1. Motivation und Überblick

1. Bedeutung der Automobilindustrie für die deutsche Volkswirtschaft

2. Motivation Automotive Software Engineering

3. GI-Fachgruppe Automotive Software Engineering

4. Software Entwicklungsprozess

5. Standards zur Softwareentwicklung

6. Software Architekturen

7. AUTOSAR

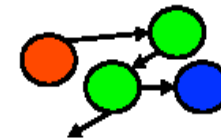
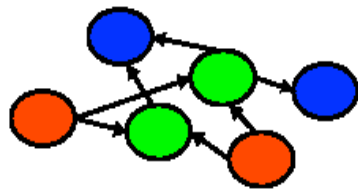
8. Lessons Learned

Software Architekturen

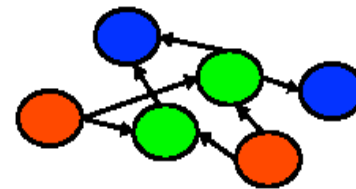
Voraussetzung für Software Wiederverwendung



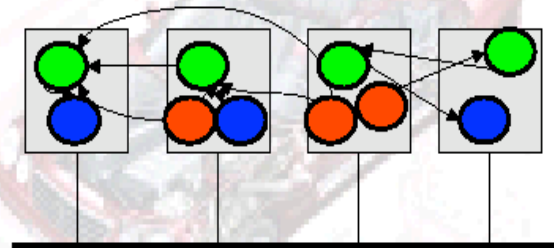
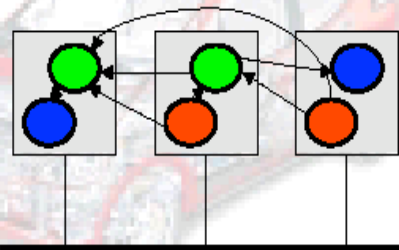
Wiederverwendung der Entwürfe
Wiederverwendung Validation and Verification
Wiederverwendung Software
Maximale Nutzung Hardware



Zusatz-
Funktionen



Basis
Funktionen

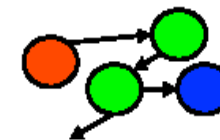
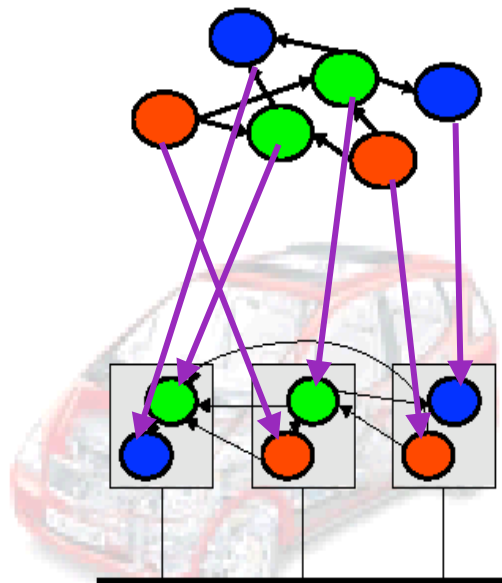


Software Architekturen

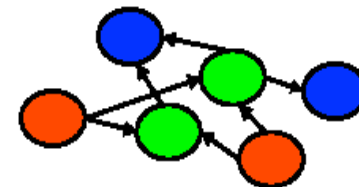
Verteilung der Funktionen auf Steuergeräte



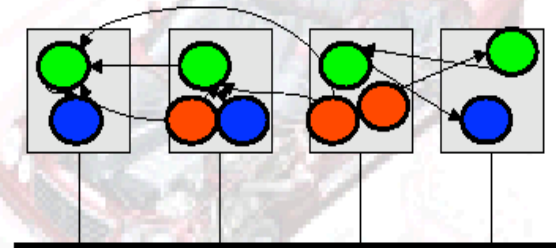
Wiederverwendung der Entwürfe
Wiederverwendung Validation and Verification
Wiederverwendung Software
Maximale Nutzung Hardware



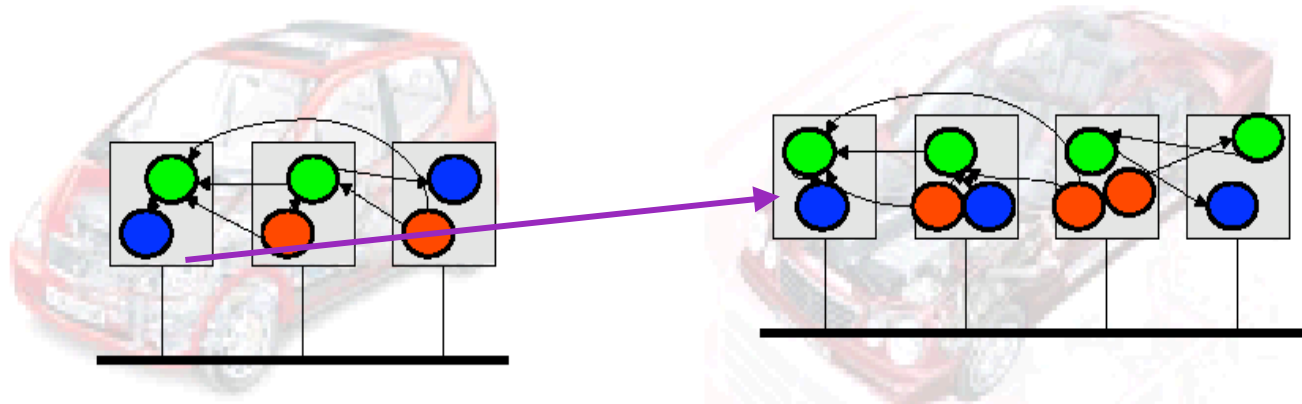
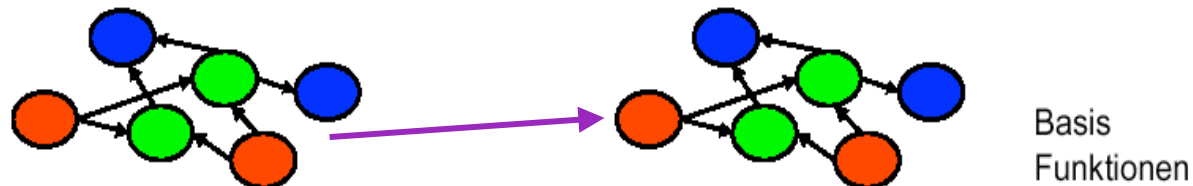
Zusatz-
Funktionen



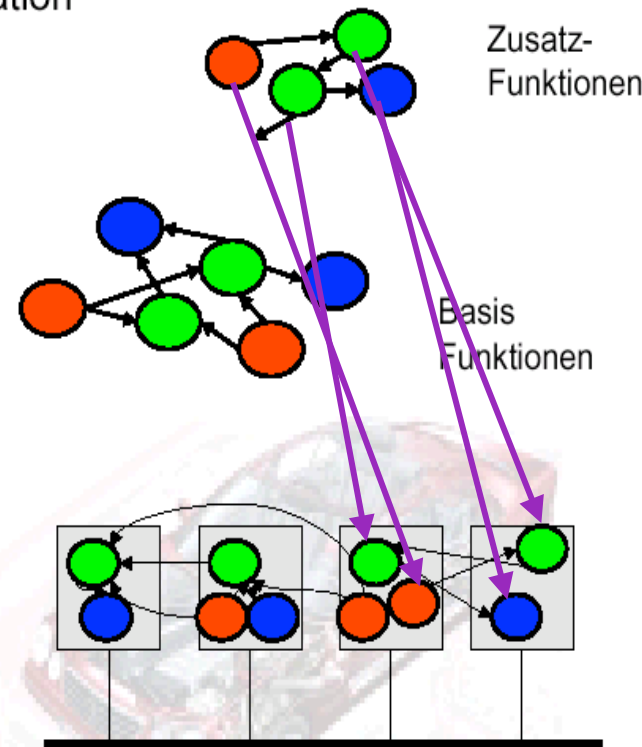
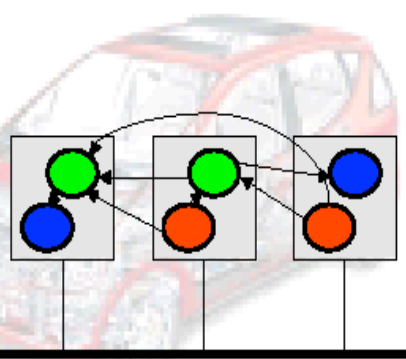
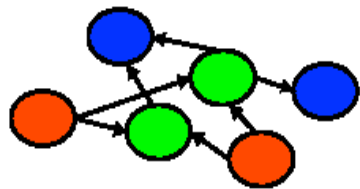
Basis
Funktionen

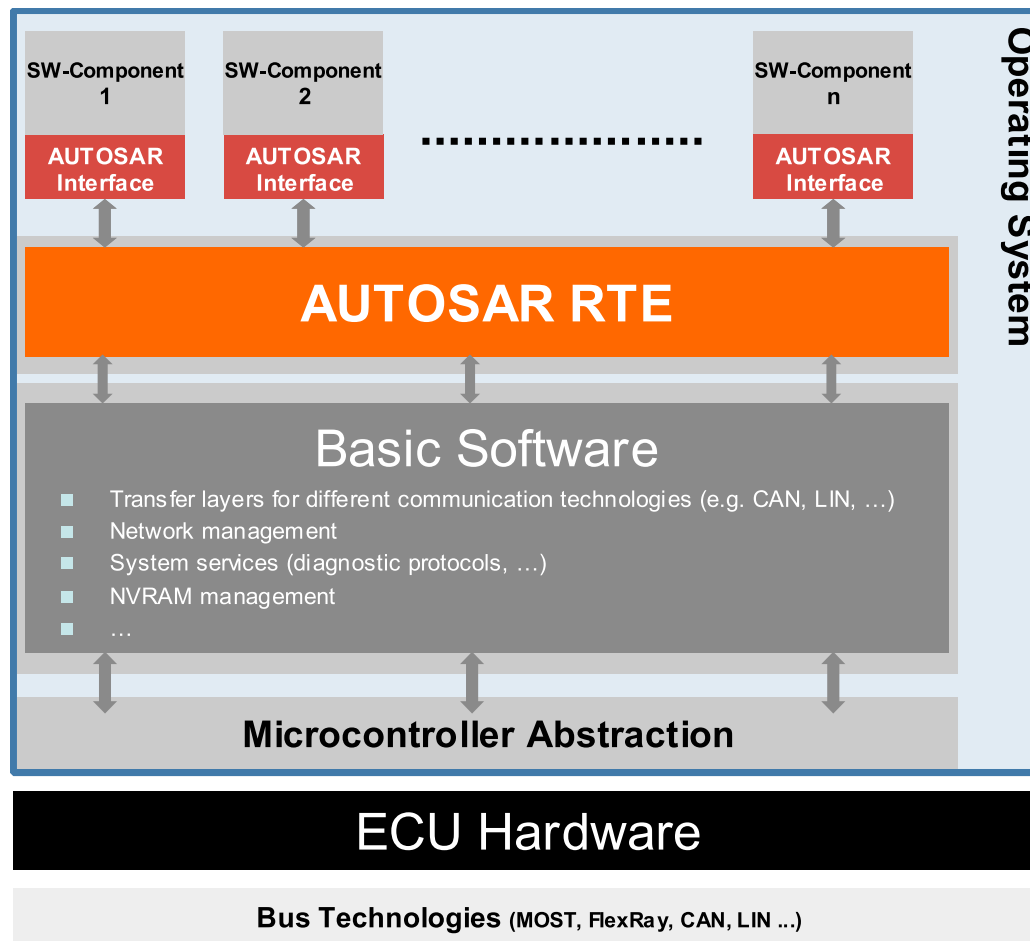


Wiederverwendung der Entwürfe
Wiederverwendung Validation and Verification
Wiederverwendung Software
Maximale Nutzung Hardware



Wiederverwendung der Entwürfe
Wiederverwendung Validation and Verification
Wiederverwendung Software
Maximale Nutzung Hardware



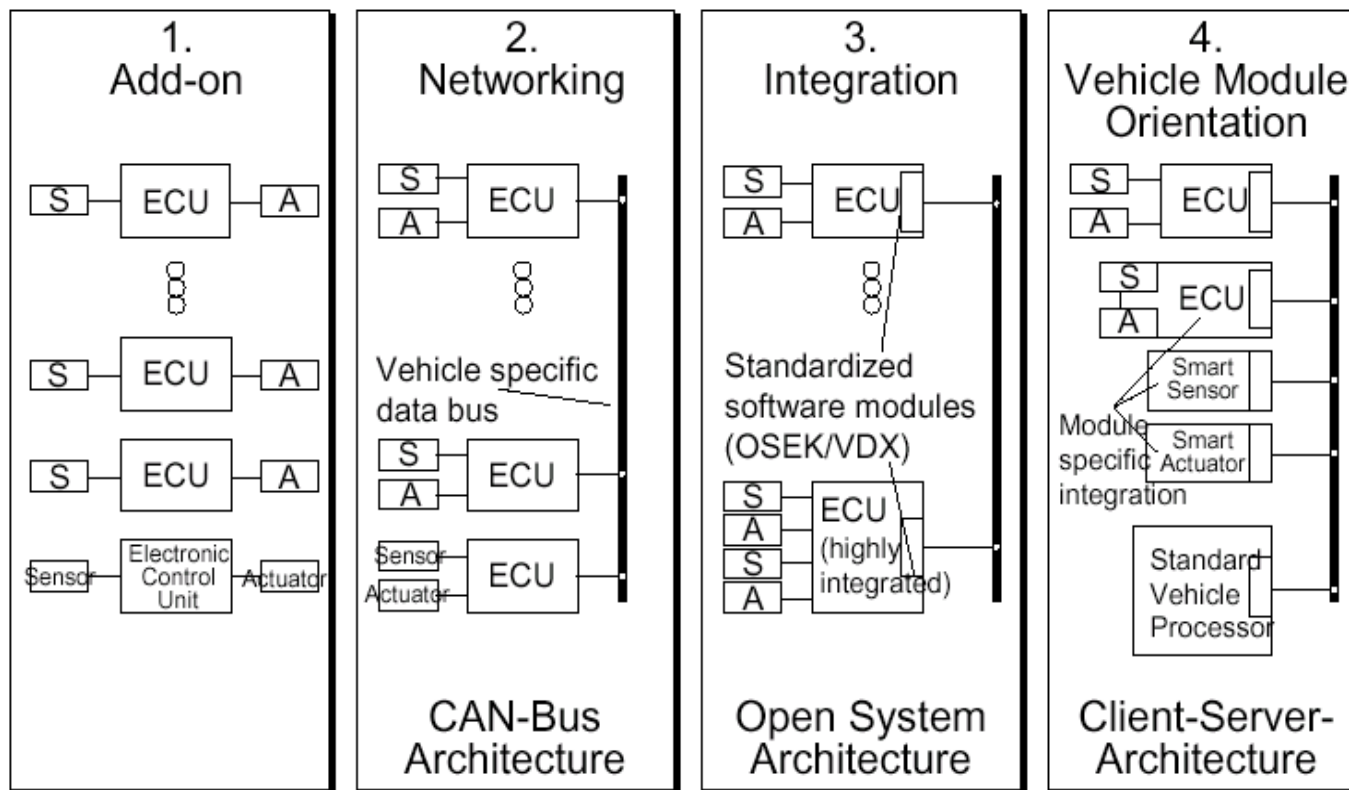


* for example: OSEK, QNX, VxWorks, Windows CE, ...

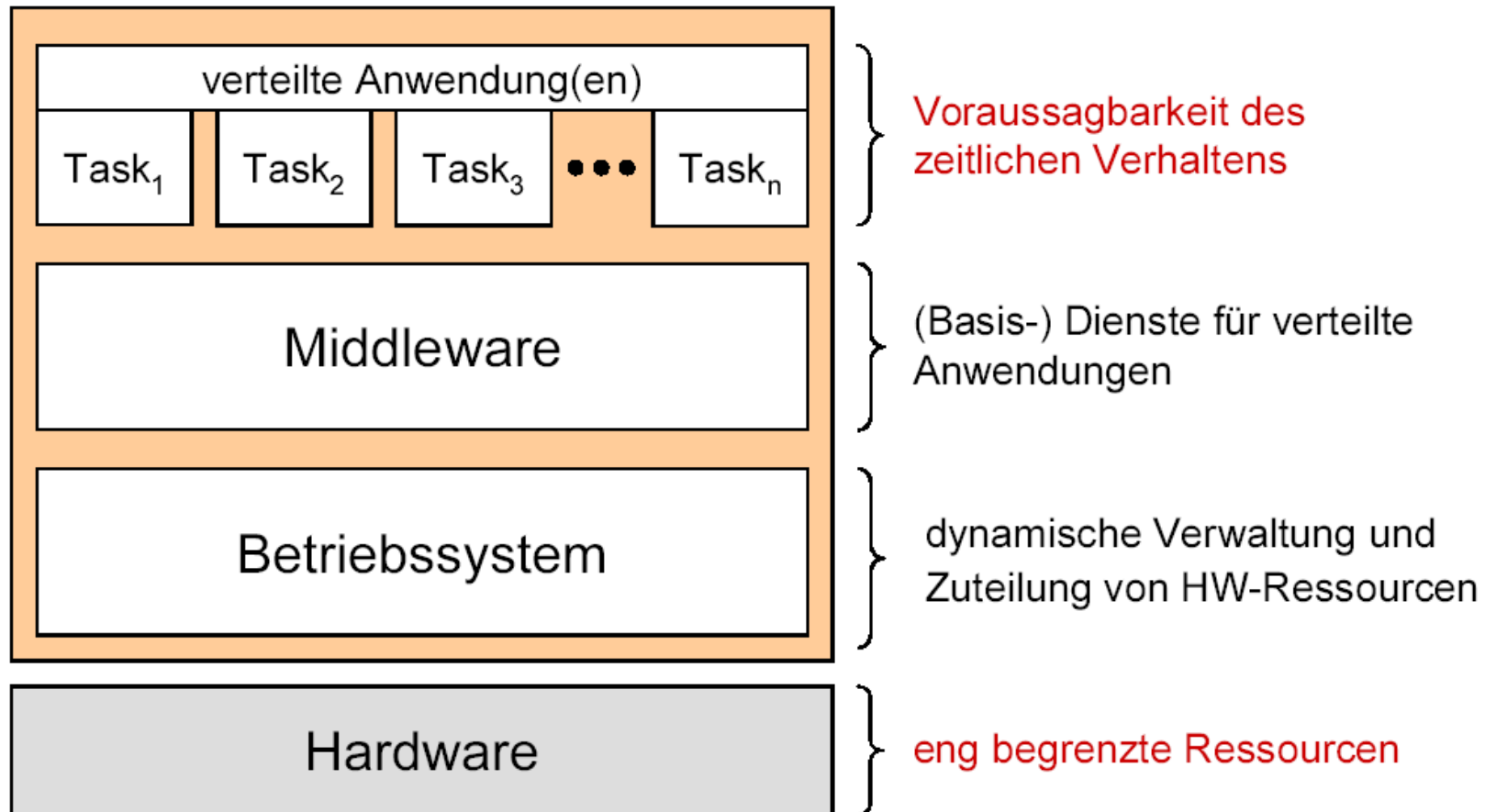
Automotive Open System Architecture:

- Standardized, openly disclosed interfaces
- HW independent SW layer
- Transferability of functions
- Redundancy activation
- **AUTOSAR RTE:**
by specifying interfaces and their communication mechanisms, the applications are decoupled from the underlying HW and Basic SW, enabling the realization of Standard Library Functions

Die Evolution der Elektrik/Elektronik Systemarchitekturen führen zu offenen Systemarchitekturen mit modularer Softwarearchitektur



Was ist Softwarearchitektur?

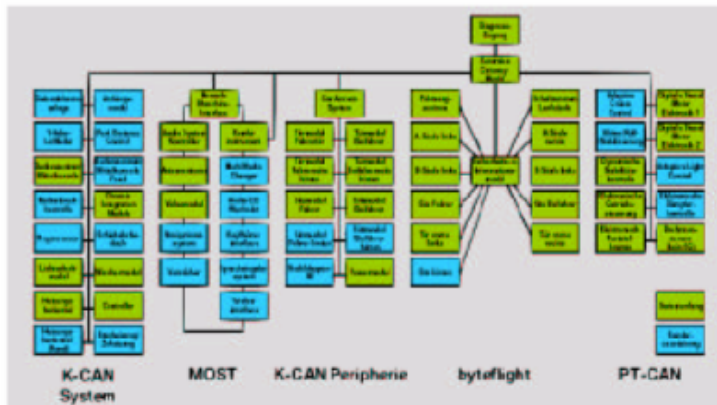


Was ist Softwarearchitektur?



- Architektur beschreibt den Aufbau eines Systems aus seinen Komponenten und deren Wechselwirkungen
- Architektur gibt Systemen Struktur und Gestalt
- Architektur strukturiert komplexe Systeme mit vielen Komponenten und macht sie überschaubar

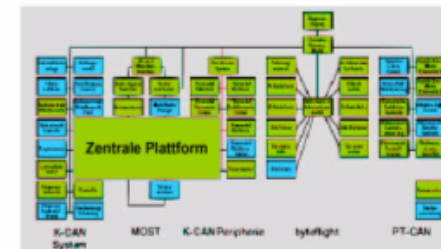
Heutige Elektroniksysteme bestehen aus hochgradig vernetzten embedded Komponenten für typischerweise jeweils eine Funktion



Zukünftig wird das bestehende System durch Zentrale Verarbeitungseinheiten ergänzt.

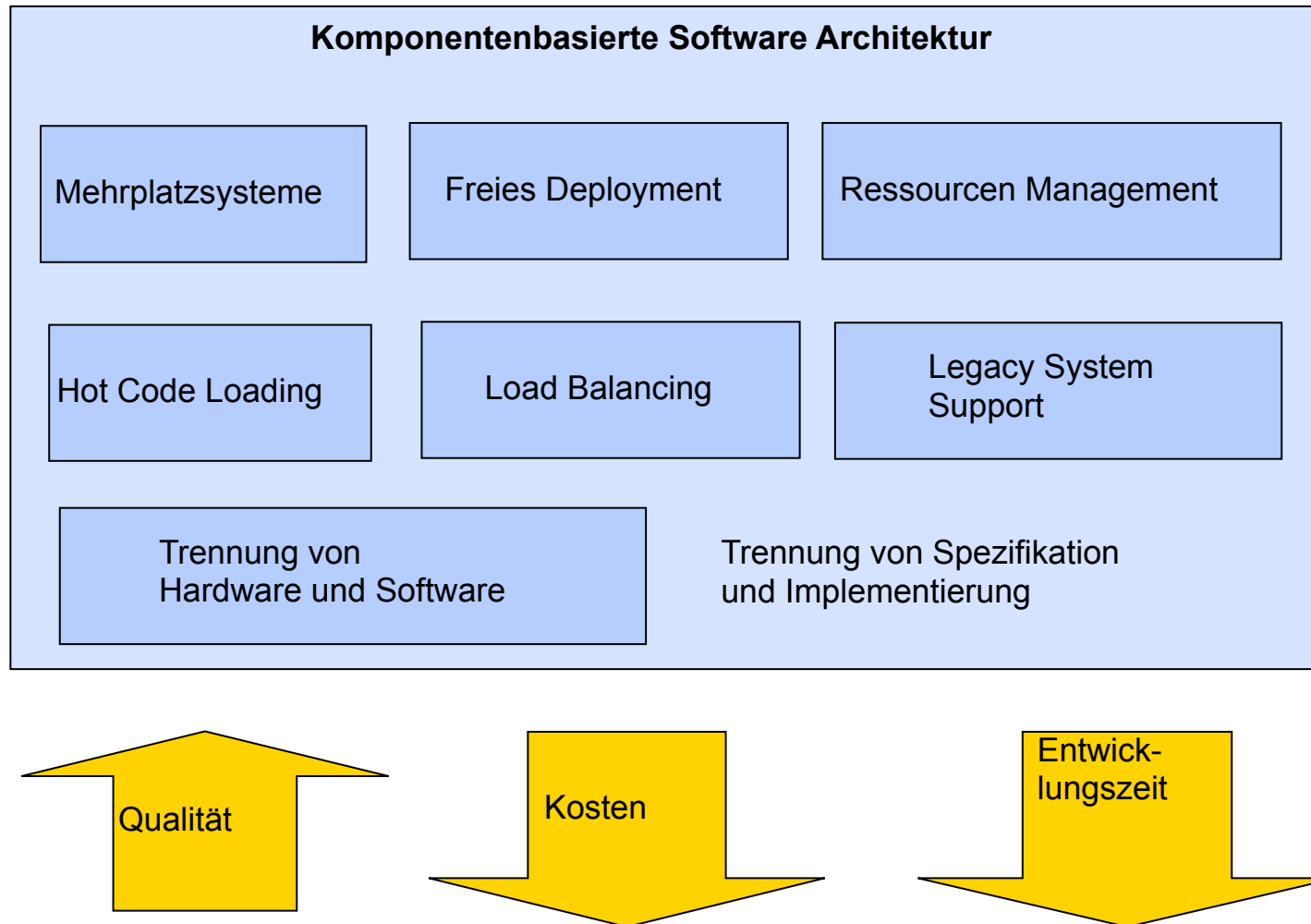
Diese Komponenten besitzen eine „offene“ Softwarearchitektur mit

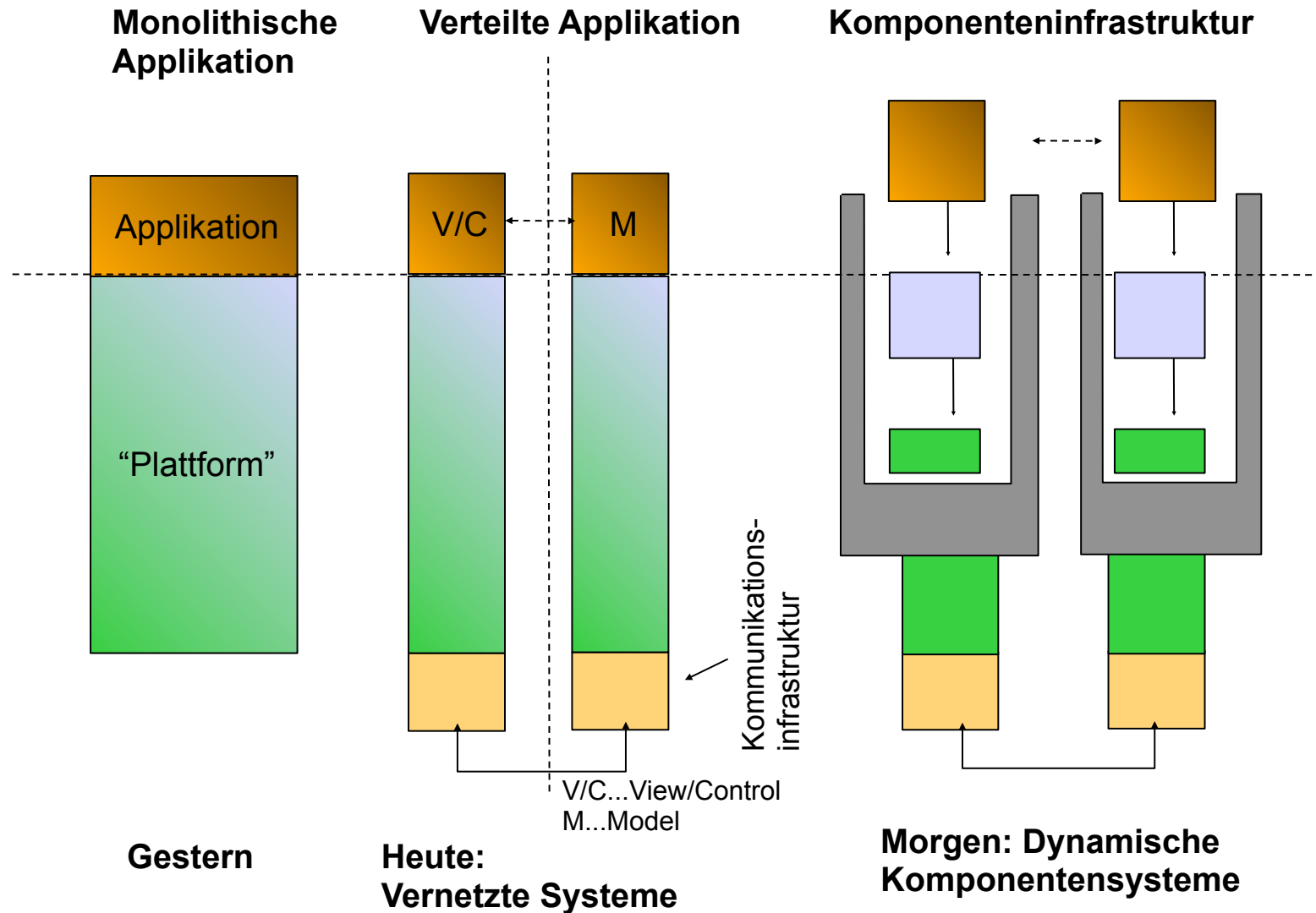
- Erweiterbarkeit und Updatefähigkeit
- Implementierung neuer Funktionalitäten
- Übergreifende Fahrzeugfunktionen

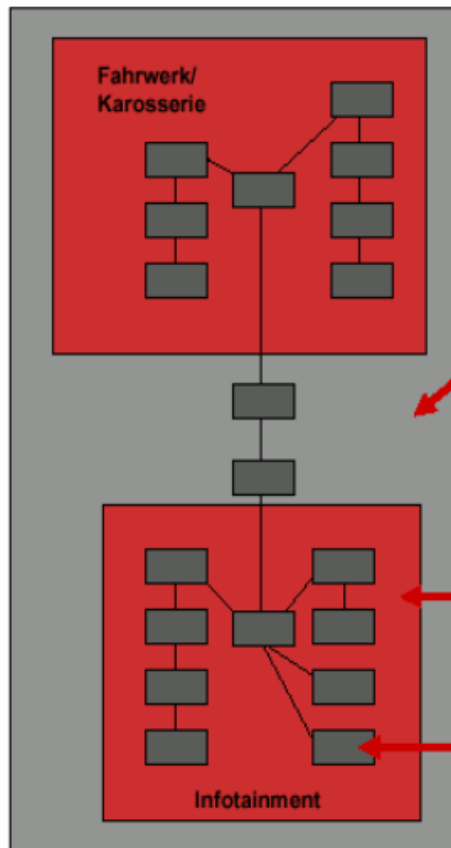


- Die Fahrzeug Software Architektur beschreibt den im Rahmen einer definierten Zielvorstellung besten Aufbau eines Fahrzeug Software Systems

- Software Architektur macht die Komplexität zukünftiger Software Systeme mit vielen wechselwirkenden Funktionen durchschaubar und beherrschbar
 - ... und begünstigt damit den Bau zuverlässiger Systeme
- Software Architektur zeigt mehrfach verwendbare Komponenten eines Software Systems auf
 - ... und schafft damit eine Grundlage für baureihenübergreifende Software Systeme
- Software Architektur zeigt Wechselwirkungen und Schnittstellen zwischen den Teilen des Software Systems auf und beschreibt diese
 - ... und schafft damit u.a. die Grundlage für die Plug&Play-Fähigkeit zukünftiger Architekturkomponenten
- Software Architektur liefert die Grundlage für
 - flexible,
 - updatefähige und
 - erweiterbare
- Software Systeme im Fahrzeug.







- **Offene Systemarchitektur**

- standardisierte, offen gelegte Schnittstellen
- hardwareunabhängige Software-Schicht
- Verschiebbare Funktionen
- Redundanzaktivierung

- **Systemarchitektur**

beschreibt die Struktur eines Systems hinsichtlich

- funktionalem Zusammenwirken und Vernetzung
- Schnittstelle zu anderen Systemen
- Daten- und Softwarearchitektur

- **System**

- funktional zusammengehöriger Verbund von Elementen

- **Element**

- z.B. Sensoren, Aktuatoren, Netzwerke, Rechnerknoten

1. Motivation und Überblick

1. Bedeutung der Automobilindustrie für die deutsche Volkswirtschaft

2. Motivation Automotive Software Engineering

3. GI-Fachgruppe Automotive Software Engineering

4. Software Entwicklungsprozess

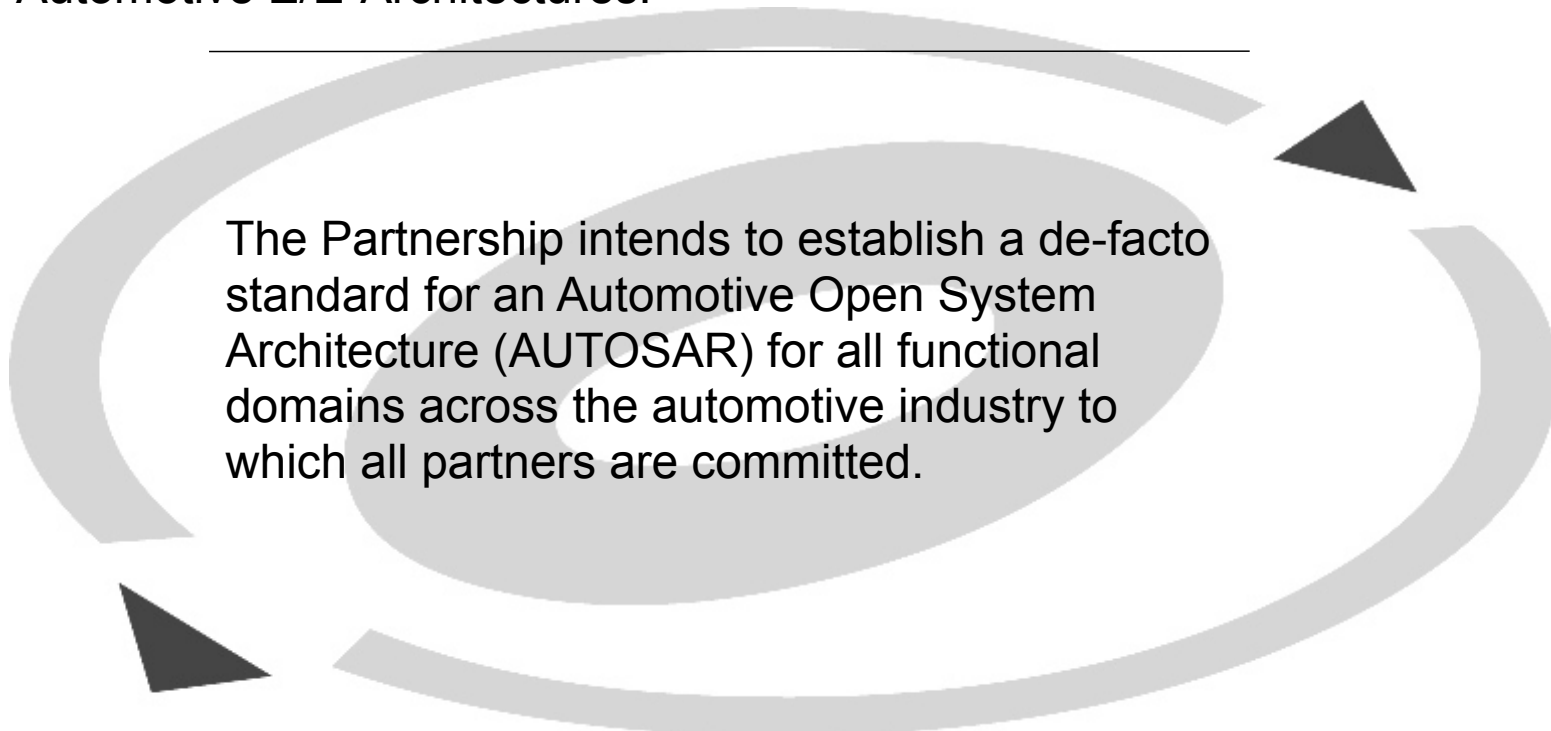
5. Standards zur Softwareentwicklung

6. Software Architekturen

7. AUTOSAR

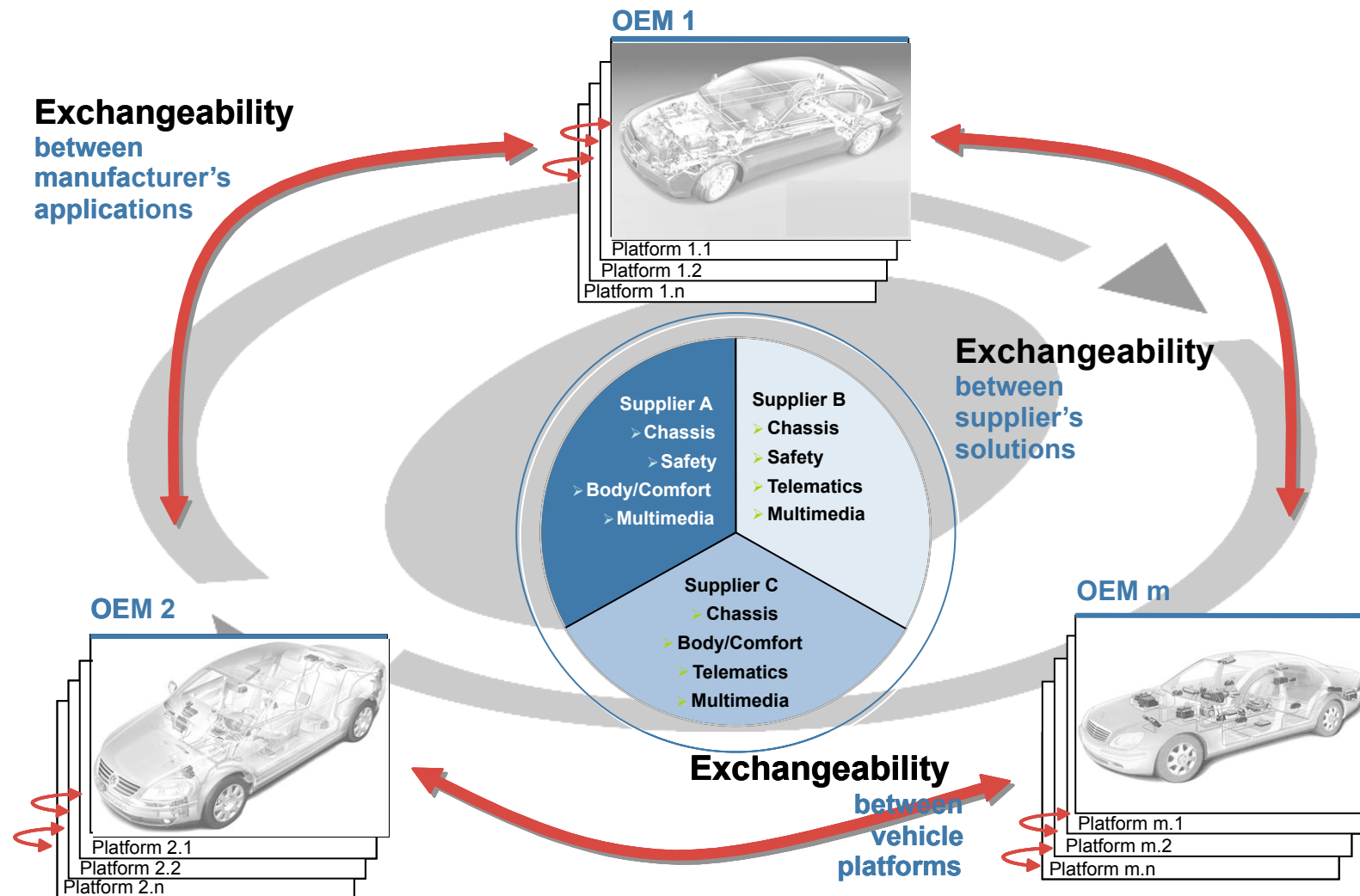
8. Lessons Learned

An industry-wide initiative to manage the complexity of emerging Automotive E/E-Architectures.

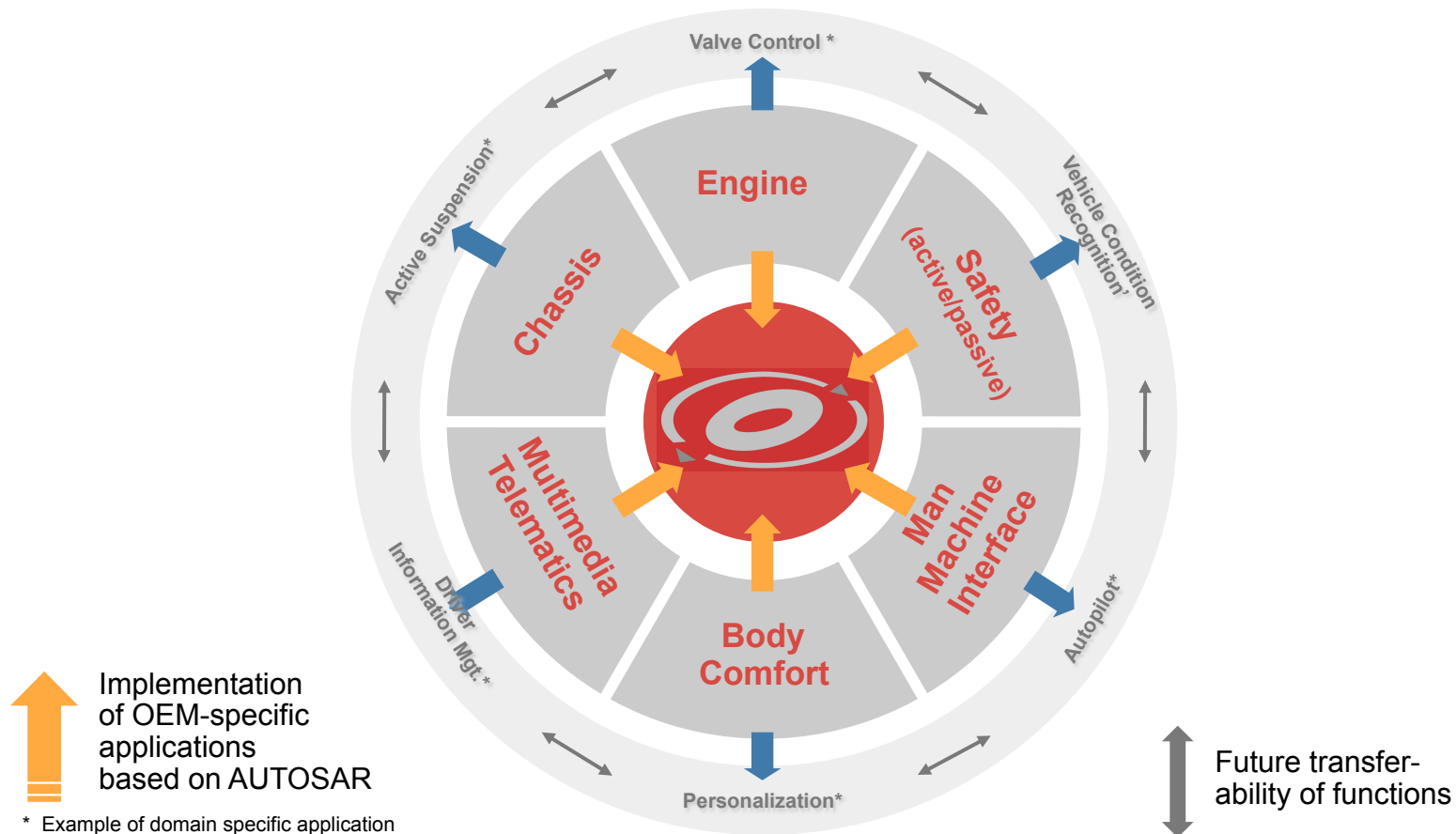


The Partnership intends to establish a de-facto standard for an Automotive Open System Architecture (AUTOSAR) for all functional domains across the automotive industry to which all partners are committed.

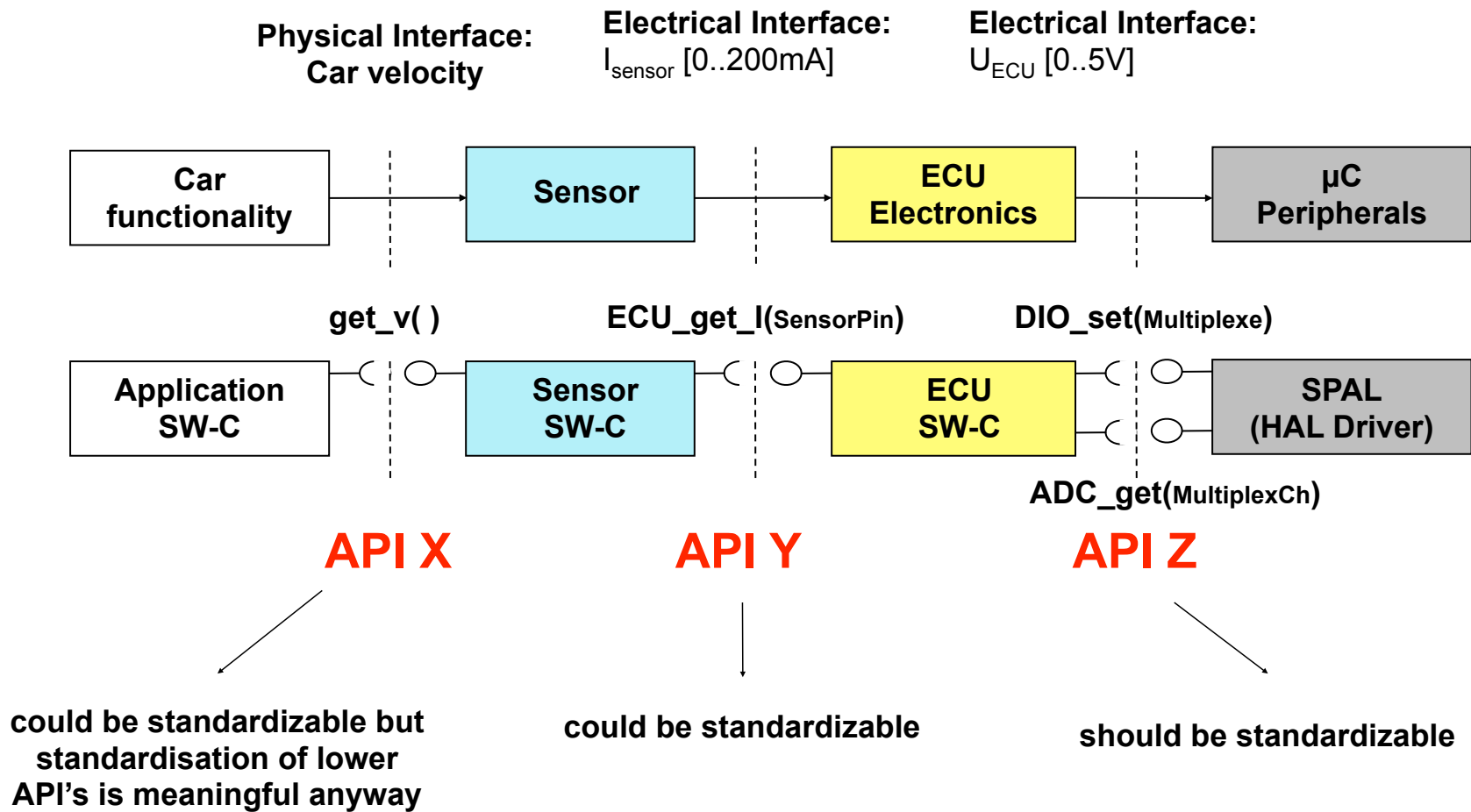


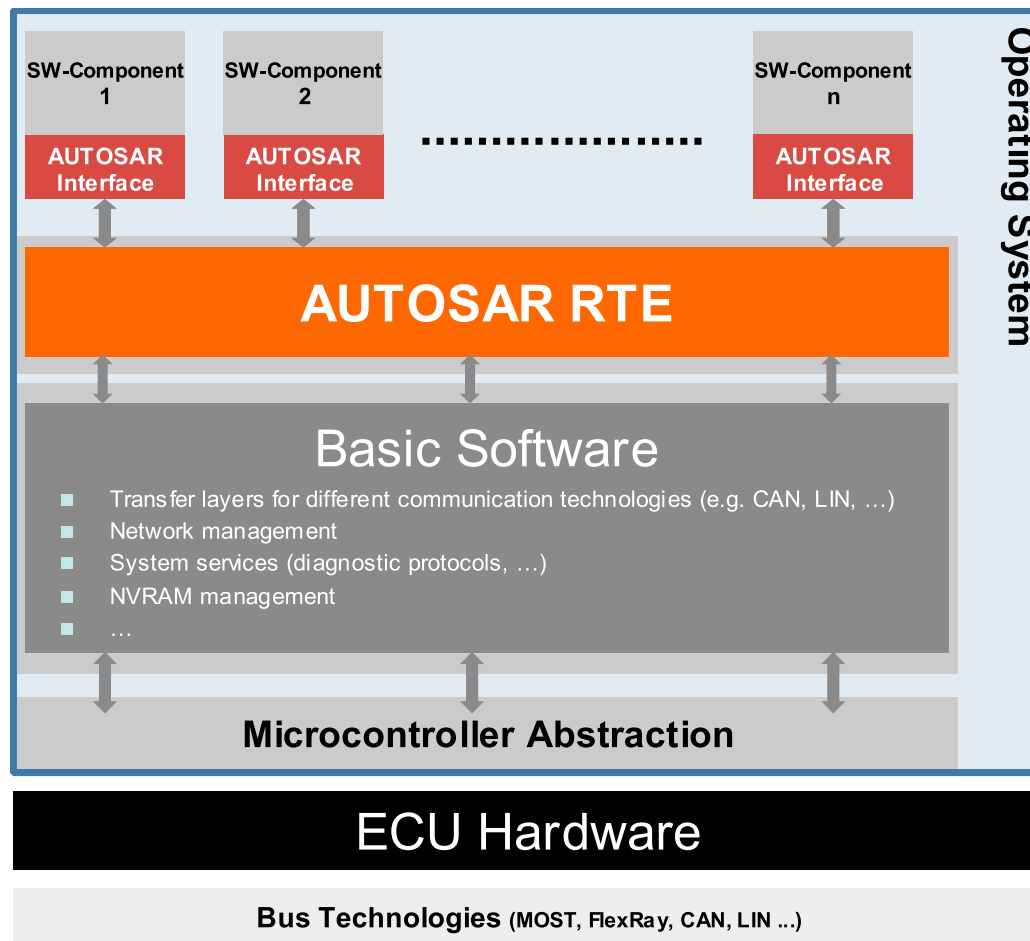


Our vision is an improved complexity management of highly integrated E/E architectures through an increased reuse and exchangeability of SW modules between OEMs and suppliers.



The primary objective of AUTOSAR is to create an infrastructure which encourages cooperation on E/E standards while maintaining competition on innovative applications.



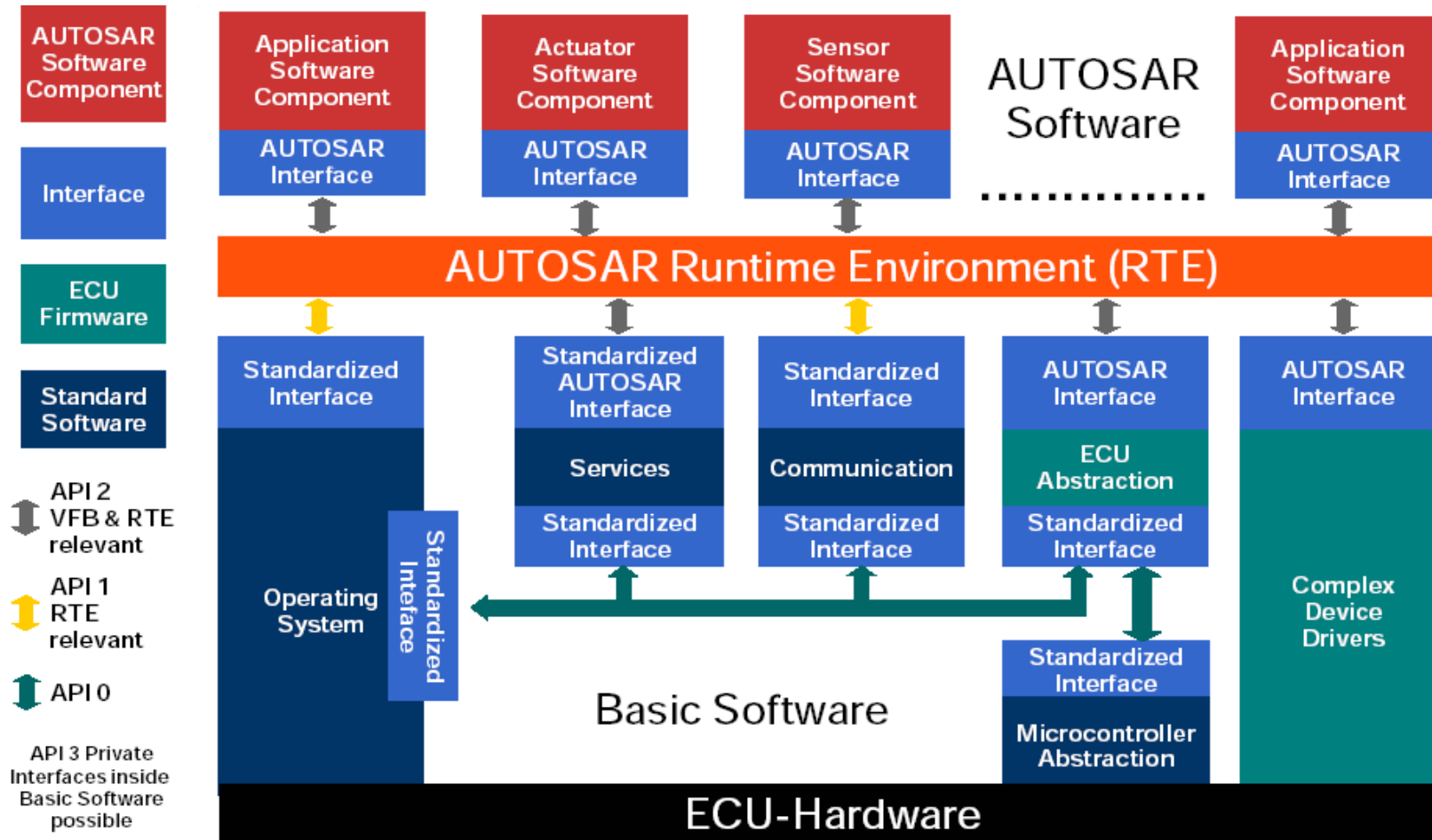


* for example: OSEK, QNX, VxWorks, Windows CE, ...

Automotive Open System Architecture:

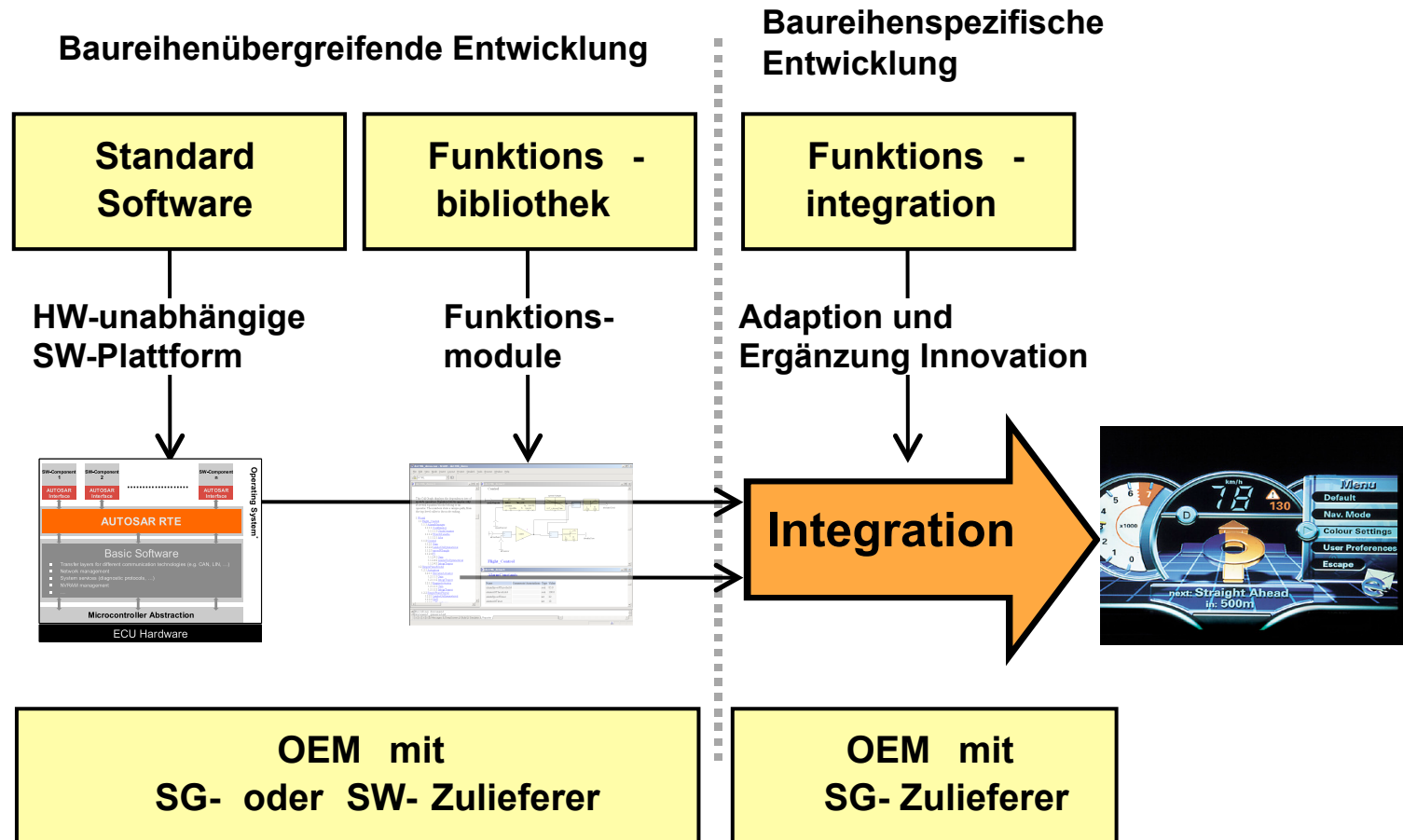
- Standardized, openly disclosed interfaces
- HW independent SW layer
- Transferability of functions
- Redundancy activation
- **AUTOSAR RTE:**
by specifying interfaces and their communication mechanisms, the applications are decoupled from the underlying HW and Basic SW, enabling the realization of Standard Library Functions

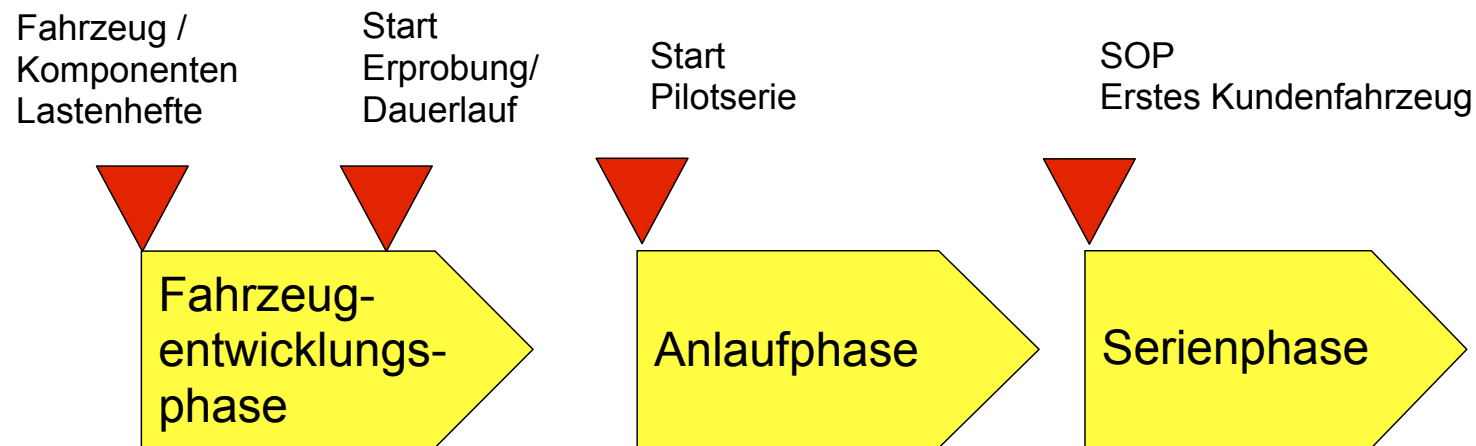
AUTOSAR Architekturkonzept



1. Motivation und Überblick

- 1. Bedeutung der Automobilindustrie für die deutsche Volkswirtschaft
- 2. Motivation Automotive Software Engineering
- 3. GI-Fachgruppe Automotive Software Engineering
- 4. Software Entwicklungsprozess
- 5. Standards zur Softwareentwicklung
- 6. Software Architekturen
- 7. AUTOSAR
- 8. Lessons Learned**





Steigende Fehlerfolgekosten „Zehner-Regel“

■ Fahrzeugentwicklungsphase

- Verzögerungen Integrationstest
- Nacharbeit im Prototypenaufbau durch Flashen bzw. Teiletausch
- Kompensation von entfallenen Erprobungsumfängen

■ Anlaufphase

- Software-Reifegradprobleme,
- Verbau nichtkompatibler SW-Versionen
- mangelndes Zusammenspiel zwischen Onboard- und Offboardsystemen
- Erhöhung der Montagezeiten sowie erhebliche Nacharbeitsumfänge (Flashen, Gerätetausch)
- Kompensation von entfallenen Erprobungs-/Dauerlaufumfängen

■ Serienphase

- Hohes Änderungsaufkommen
- Kundenunzufriedenheit
- Rückrufe

■ Produkthaftungsfälle

Top 5 Risiken Embedded Software



- Optimierte Verteilung und Vernetzung von Systemen im Fahrzeug
- Funktionsspezifikation verteilter Systeme
- Qualitäts- und kostenorientierte Kooperation mit n Lieferanten für m Module
- Gestaltung des Entwicklungs- und Integrationsprozesses
- Software-Updates/Upgrades im Feld

Reihenfolge Top 3 Maßnahmen



■ Ordnung schaffen

- Software-Entwicklungsprozesse mit Fahrzeug- und SG-Entwicklungsprozessen koppeln
- Releasemanagement bezüglich der Einsteuerung von SW-Versionen in Fahrzeug-Baulose
- Konfigurationsmanagement bei Software

■ Änderungswesen organisieren

- Änderungsmanagement fahrzeugseitig
- Änderungsmanagement lieferantenseitig
- Änderungsstopps klar kommunizieren, ggf. eskalieren

■ Transparenz schaffen

- Objektive Meßgrößen zur Bewertung des SW-Reifegrades einführen
- Anflugkurven zur Beschreibung der Erwartungswerte definieren
- Nicht nur das Fehleraufkommen, auch die Testabdeckung, die Fehlerbehebungszeiten bzw. die Zuverlässigkeit der Fehlerbehebung liefern wichtige Erkenntnisse

- Durch steigende Komplexität wird Software-Qualitäts-und Prozeßmanagement zum elementaren Bestandteil des Fahrzeugentwicklungsprozesses.
- Die Schwerpunkte des Fahrzeugherstellers liegen hierbei auf den Spezifikations-, Integrations- und Testschritten
- Grundlegende Voraussetzungen für die effiziente Durchführung sind:
 - Synchronisation der SW-Entwicklungsstände aller Steuergeräte mit den Meilensteinen und Freigabeprozessen der Fahrzeugentwicklung.
 - Konsequente Durchführung Änderungsmanagement.
 - Bewertung und Steuerung des Software-Reifegrades auf Basis eines Reifegradbewertungsprozesses mit objektiven Meßgrößen.
- Wichtige Grundlage ist die Standardisierung nicht wettbewerbsrelevanter Software Umfänge.

